INT-353

Name : IMMIDICHETTY REDDY SWETHAK

Reg.no : 12008336

Section : K20MP

Roll.no : RK20MPA18

Dataset : NBA Games Data

Introduction :

This dataset was selected to work on the NBA games data. I taken the data from the nba stats website to create this dataset. In this dataset the data is consists from the 2004 season to December 2020 season.

In [1]:
```
# Importing the libraries
import numpy as np
import pandas as pd
```

In [2]:
```
#Reading csv file
df = pd.read_csv("C:/Users/swethak/Desktop/EDA project/games.csv")
```

In [3]:
```
# Display the data set
df
```

Out[3]:

	GAME_DATE_EST	GAME_ID	GAME_STATUS_TEXT	HOME_TEAM_ID	VISITOR_TEAM_ID	SEAS
0	2022-03-12	22101005	Final	1610612748	1610612750	2
1	2022-03-12	22101006	Final	1610612741	1610612739	2
2	2022-03-12	22101007	Final	1610612759	1610612754	2
3	2022-03-12	22101008	Final	1610612744	1610612749	2
4	2022-03-12	22101009	Final	1610612743	1610612761	2
...	...	...	...	...	...	
25791	2014-10-06	11400007	Final	1610612737	1610612740	2
25792	2014-10-06	11400004	Final	1610612741	1610612764	2
25793	2014-10-06	11400005	Final	1610612747	1610612743	2
25794	2014-10-05	11400002	Final	1610612761	1610612758	2
25795	2014-10-04	11400001	Final	1610612748	1610612740	2

25796 rows × 21 columns

In [4]:
```
# Display first 5 rows
df.head()
```

Out[4]:

	GAME_DATE_EST	GAME_ID	GAME_STATUS_TEXT	HOME_TEAM_ID	VISITOR_TEAM_ID	SEASON
0	2022-03-12	22101005	Final	1610612748	1610612750	2021
1	2022-03-12	22101006	Final	1610612741	1610612739	2021
2	2022-03-12	22101007	Final	1610612759	1610612754	2021
3	2022-03-12	22101008	Final	1610612744	1610612749	2021
4	2022-03-12	22101009	Final	1610612743	1610612761	2021

5 rows × 21 columns

In [5]:
```
# Display last 5 rows
df.tail()
```

Out[5]:

	GAME_DATE_EST	GAME_ID	GAME_STATUS_TEXT	HOME_TEAM_ID	VISITOR_TEAM_ID	SEAS
25791	2014-10-06	11400007	Final	1610612737	1610612740	2
25792	2014-10-06	11400004	Final	1610612741	1610612764	2
25793	2014-10-06	11400005	Final	1610612747	1610612743	2
25794	2014-10-05	11400002	Final	1610612761	1610612758	2
25795	2014-10-04	11400001	Final	1610612748	1610612740	2

5 rows × 21 columns

In [6]:
```
# Information about the dataset
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 25796 entries, 0 to 25795
Data columns (total 21 columns):
 #   Column            Non-Null Count  Dtype  
---  ------            --------------  -----  
 0   GAME_DATE_EST     25796 non-null  object 
 1   GAME_ID           25796 non-null  int64  
 2   GAME_STATUS_TEXT  25796 non-null  object 
 3   HOME_TEAM_ID      25796 non-null  int64  
 4   VISITOR_TEAM_ID   25796 non-null  int64  
 5   SEASON            25796 non-null  int64  
 6   TEAM_ID_home      25796 non-null  int64  
 7   PTS_home          25697 non-null  float64
 8   FG_PCT_home       25697 non-null  float64
 9   FT_PCT_home       25697 non-null  float64
 10  FG3_PCT_home      25697 non-null  float64
 11  AST_home          25697 non-null  float64
 12  REB_home          25697 non-null  float64
 13  TEAM_ID_away      25796 non-null  int64  
 14  PTS_away          25697 non-null  float64
 15  FG_PCT_away       25697 non-null  float64
 16  FT_PCT_away       25697 non-null  float64
 17  FG3_PCT_away      25697 non-null  float64
 18  AST_away          25697 non-null  float64
 19  REB_away          25697 non-null  float64
 20  HOME_TEAM_WINS    25796 non-null  int64  
dtypes: float64(12), int64(7), object(2)
memory usage: 4.1+ MB
```

In [7]:
```
# Shape of the dataset
df.shape
```

Out[7]:
```
(25796, 21)
```

In [8]:
```
#Size of the dataset
df.size
```

Out[8]:
```
541716
```

In [9]:
```
#Describe the dataset
df.describe()
```

Out[9]:

	GAME_ID	HOME_TEAM_ID	VISITOR_TEAM_ID	SEASON	TEAM_ID_home	PTS_ho
count	2.579600e+04	2.579600e+04	2.579600e+04	25796.000000	2.579600e+04	25697.000
mean	2.169208e+07	1.610613e+09	1.610613e+09	2011.798341	1.610613e+09	103.106
std	5.496041e+06	8.638857e+00	8.654846e+00	5.397985	8.638857e+00	13.174
min	1.030000e+07	1.610613e+09	1.610613e+09	2003.000000	1.610613e+09	36.000
25%	2.060109e+07	1.610613e+09	1.610613e+09	2007.000000	1.610613e+09	94.000
50%	2.120040e+07	1.610613e+09	1.610613e+09	2012.000000	1.610613e+09	103.000
75%	2.170070e+07	1.610613e+09	1.610613e+09	2016.000000	1.610613e+09	112.000
max	5.200021e+07	1.610613e+09	1.610613e+09	2021.000000	1.610613e+09	168.000

In [10]:
```
#knowing the datatypes
df.dtypes
```

Out[10]:
```
GAME_DATE_EST        object
GAME_ID               int64
GAME_STATUS_TEXT     object
HOME_TEAM_ID          int64
VISITOR_TEAM_ID       int64
SEASON                int64
TEAM_ID_home          int64
PTS_home            float64
FG_PCT_home         float64
FT_PCT_home         float64
FG3_PCT_home        float64
AST_home            float64
REB_home            float64
TEAM_ID_away          int64
PTS_away            float64
FG_PCT_away         float64
FT_PCT_away         float64
FG3_PCT_away        float64
AST_away            float64
REB_away            float64
HOME_TEAM_WINS        int64
dtype: object
```

Data handling and cleaning

In [11]:
```
#drop duplicates
df.drop_duplicates(subset=['GAME_STATUS_TEXT'])
```

Out[11]:

	GAME_DATE_EST	GAME_ID	GAME_STATUS_TEXT	HOME_TEAM_ID	VISITOR_TEAM_ID	SEASON
0	2022-03-12	22101005	Final	1610612748	1610612750	2021

1 rows × 21 columns

In [12]:
```
#checking the null values
df.isnull().sum()
```

Out[12]:
```
GAME_DATE_EST        0
GAME_ID              0
GAME_STATUS_TEXT     0
HOME_TEAM_ID         0
VISITOR_TEAM_ID      0
SEASON               0
TEAM_ID_home         0
PTS_home            99
FG_PCT_home         99
FT_PCT_home         99
FG3_PCT_home        99
AST_home            99
REB_home            99
TEAM_ID_away         0
PTS_away            99
FG_PCT_away         99
FT_PCT_away         99
FG3_PCT_away        99
AST_away            99
REB_away            99
HOME_TEAM_WINS       0
dtype: int64
```

In [13]:
```
#replacing missing values
nr=df['PTS_home'].replace(np.NaN,df['PTS_home'].median(),inplace=True)
nr
```

In [14]:
```
#checking the null values
df.isnull().sum()
```

Out[14]:
```
GAME_DATE_EST        0
GAME_ID              0
GAME_STATUS_TEXT     0
HOME_TEAM_ID         0
VISITOR_TEAM_ID      0
SEASON               0
TEAM_ID_home         0
PTS_home             0
FG_PCT_home         99
FT_PCT_home         99
FG3_PCT_home        99
AST_home            99
REB_home            99
TEAM_ID_away         0
PTS_away            99
FG_PCT_away         99
FT_PCT_away         99
FG3_PCT_away        99
AST_away            99
REB_away            99
HOME_TEAM_WINS       0
dtype: int64
```

In [15]:
```
#drops the null values in all the columns
df=df.dropna(subset=["FG_PCT_home","FG3_PCT_away","REB_home","FG3_PCT_home","AST_h
df.isnull().sum()
```

Out[15]:
```
GAME_DATE_EST       0
GAME_ID             0
GAME_STATUS_TEXT    0
HOME_TEAM_ID        0
VISITOR_TEAM_ID     0
SEASON              0
TEAM_ID_home        0
PTS_home            0
FG_PCT_home         0
FT_PCT_home         0
FG3_PCT_home        0
AST_home            0
REB_home            0
TEAM_ID_away        0
PTS_away            0
FG_PCT_away         0
FT_PCT_away         0
FG3_PCT_away        0
AST_away            0
REB_away            0
HOME_TEAM_WINS      0
dtype: int64
```

Inferential Statistics

In [16]:
```
import seaborn as sns
```

In [17]:
```
#knowing the dtype of the given column
df['FG3_PCT_away'].mode()
```

Out[17]:
```
0    0.333
Name: FG3_PCT_away, dtype: float64
```

In [18]:
```
#knowing the mean of the given column
df['FG3_PCT_away'].mean()
```

Out[18]:
```
0.34941312215433046
```

In [19]:
```
#knowing the mode of the given column
df['FG3_PCT_away'].mode()
```

Out[19]:
```
0    0.333
Name: FG3_PCT_away, dtype: float64
```

In [20]:
```
#knowing the median of the given column
df['FG3_PCT_away'].median()
```

Out[20]:
```
0.35
```

In [21]:
```
from sklearn.impute import SimpleImputer
```

In [25]:
```
#detecting position of outliers
print(np.where(df['FG3_PCT_away']>0))
```

```
(array([    0,     1,     2, ..., 25694, 25695, 25696], dtype=int64),)
```

In [26]:
```
#detection of outliers using z-zscore method
from scipy import stats
import numpy as np
z=np.abs(stats.zscore(df['FG3_PCT_away']))
print(z)
```

```
0        0.068852
1        1.283337
2        0.359255
3        0.332029
4        0.341104
           ...
25791    0.232203
25792    0.747907
25793    1.366590
25794    0.322954
25795    0.803934
Name: FG3_PCT_away, Length: 25697, dtype: float64
```

In [27]:
```
#detecting outliers using IQR method
Q1=np.percentile(df['FG3_PCT_away'],25,interpolation = 'midpoint')
Q3=np.percentile(df['FG3_PCT_away'],75,interpolation = 'midpoint')
IQR = Q3-Q1
```

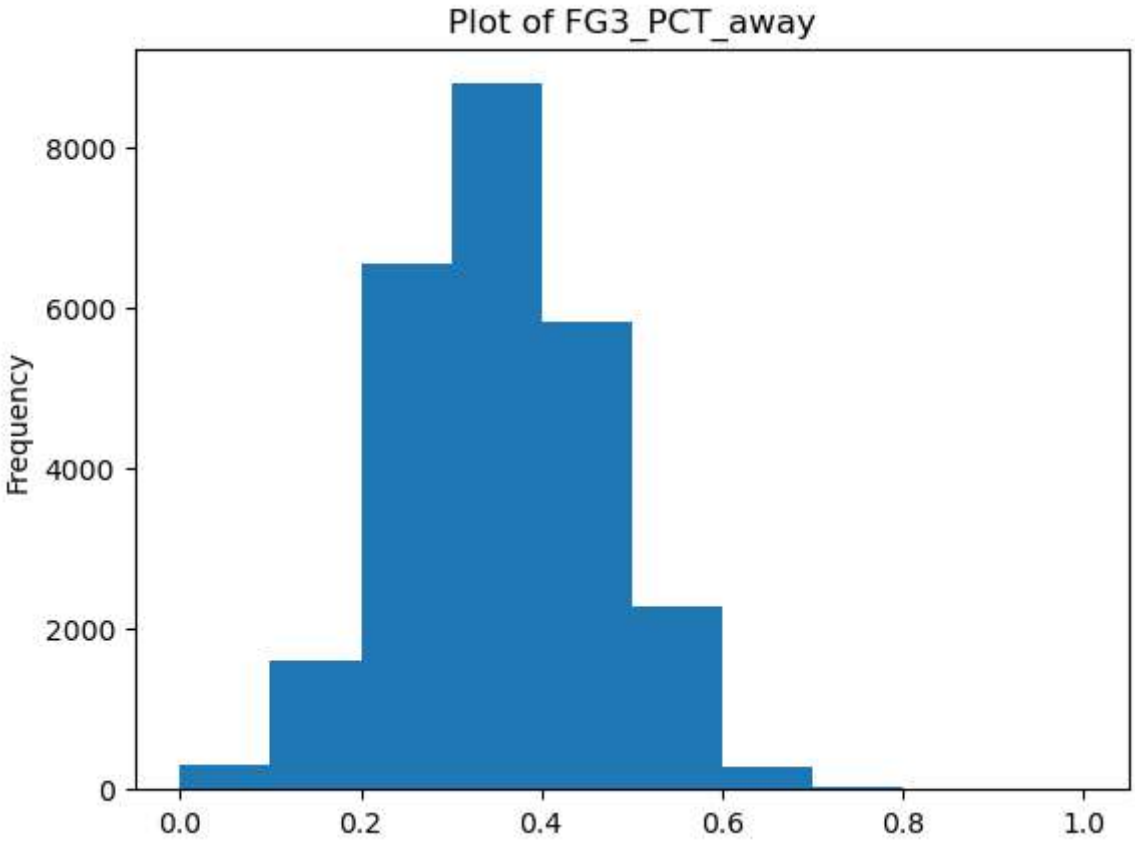
In [28]:
```
upper = df['FG3_PCT_away'] >=(Q3+1.5*IQR)
print("Upper bound:",upper)
print(np.where(upper))

lower = df['FG3_PCT_away'] <= (Q1-1.5*IQR)
print("Lower bound:",lower)
print(np.where(lower))
```

```
Upper bound: 0        False
1        False
2        False
3        False
4        False
         ...
25791    False
25792    False
25793    False
25794    False
25795    False
Name: FG3_PCT_away, Length: 25697, dtype: bool
(array([ 1298,  2607,  3898,  3961,  4499,  4554,  4623,  5162,  5276,
        5452,  6081,  6268,  6381,  6440,  6512,  6563,  6672,  6904,
        7034,  7497,  8110,  8121,  8136,  8185,  8229,  8285,  8314,
        8357,  8399,  8424,  8434,  8568,  8581,  8760,  8777,  8950,
        8997,  9094,  9157,  9185,  9203,  9244,  9353,  9403,  9671,
        9775,  9788,  9831,  9913,  9943,  9954,  9968, 10140, 10227,
       10242, 10334, 10421, 10510, 10563, 10593, 10743, 10766, 10854,
       10940, 11043, 11120, 11149, 11177, 11265, 11386, 11445, 11734,
       11797, 11855, 11915, 11954, 11966, 11971, 11987, 12005, 12085,
       12123, 12186, 12193, 12254, 12646, 12765, 13013, 13031, 13145,
       13285, 13364, 13430, 13544, 13547, 13698, 13718, 13729, 13744,
       13772, 13879, 13882, 14089, 14116, 14519, 14562, 14618, 14678,
       14689, 14703, 14745, 14753, 14761, 14792, 14818, 14833, 14864,
       14892, 14922, 14980, 14983, 15018, 15162, 15201, 15252, 15272,
       15281, 15282, 15325, 15339, 15353, 15367, 15485, 15514, 15680,
       15882, 15893, 15937, 16051, 16098, 16130, 16162, 16171, 16216,
       16228, 16271, 16314, 16362, 16380, 16442, 16468, 16586, 16620,
       16710, 16842, 16851, 16883, 16967, 17229, 17251, 17337, 17366,
       17493, 17569, 17662, 17663, 17669, 17684, 17700, 17706, 17751,
       17814, 17849, 17921, 17992, 18008, 18045, 18070, 18075, 18168,
       18214, 18259, 18323, 18327, 18610, 19207, 19522, 20352, 20595,
       20857, 21731, 22485, 22899, 23226, 23539, 23965, 24068, 24945,
       25208, 25244], dtype=int64),)
Lower bound: 0        False
1        False
2        False
3        False
4        False
         ...
25791    False
25792    False
25793    False
25794    False
25795    False
Name: FG3_PCT_away, Length: 25697, dtype: bool
(array([ 3337,  3487,  4525,  4884,  4925,  5347,  5567,  5609,  6128,
        6198,  6509,  6526,  6544,  6634,  6841,  6873,  7016,  7028,
        7030,  7066,  7116,  7127,  7596,  7699,  7854,  7856,  8286,
        8385,  8419,  8515,  8867,  8922,  8981,  9098,  9152,  9267,
        9283,  9418,  9438,  9769,  9875,  9878, 10105, 10232, 10333,
       10686, 10899, 11273, 11285, 11308, 11414, 11496, 11800, 11904,
       11936, 12002, 12010, 12214, 12408, 12413, 12568, 12680, 12720,
       12787, 12795, 12953, 12973, 13052, 13074, 13137, 13209, 13245,
       13272, 13713, 14006, 14027, 14033, 14046, 14158, 14220, 14274,
       14634, 14730, 14795, 14803, 14851, 14889, 15129, 15208, 15215,
       15247, 15307, 15528, 15557, 15572, 15579, 15596, 15608, 15609,
       15626, 15656, 15692, 15857, 15884, 16016, 16351, 16365, 16381,
       16429, 16533, 16558, 16657, 16658, 16679, 16728, 16816, 16826,
       16895, 16923, 16970, 17037, 17038, 17083, 17247, 17267, 17320,
       17325, 17330, 17495, 17876, 17985, 17998, 18023, 18182, 18297,
       18324, 22525, 22832, 22933, 23422, 23486, 23865, 24885, 25183,
       25455, 25559], dtype=int64),)
```

Univariate Analysis

In [29]:
```
import matplotlib.pyplot as plt
```

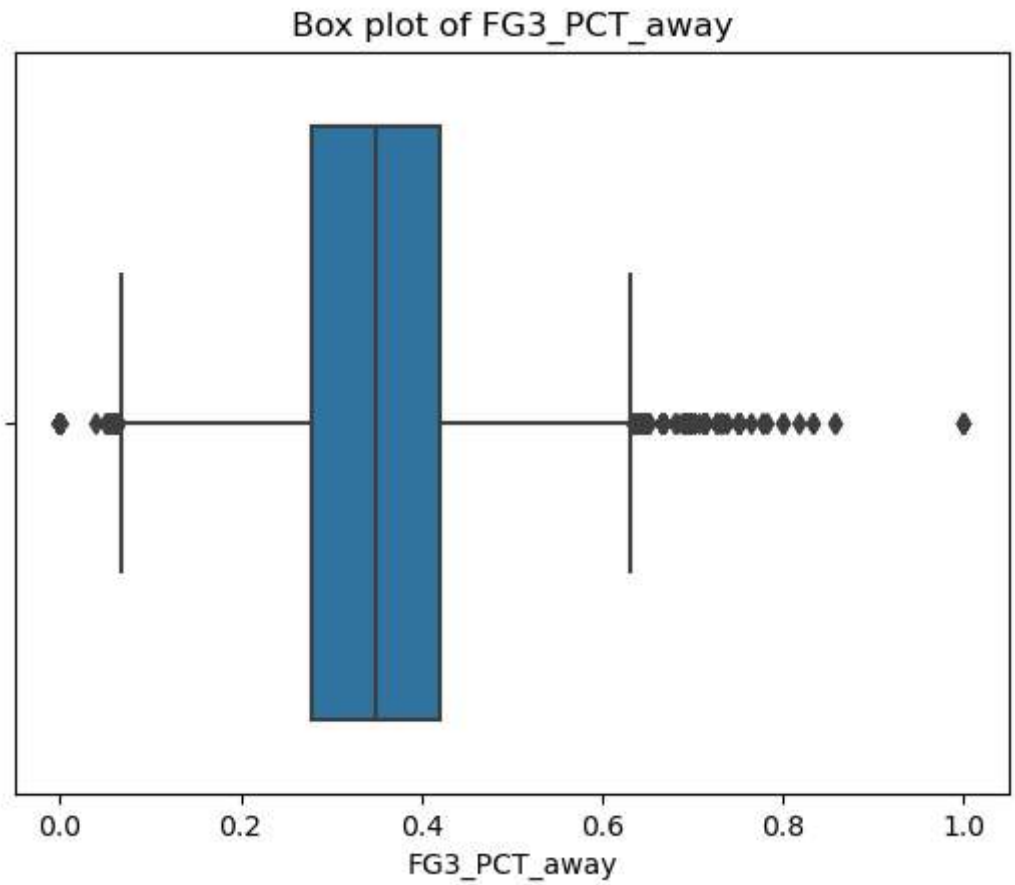
In [30]:
```
df.FG3_PCT_away.plot.hist()
plt.title("Plot of FG3_PCT_away")
plt.show()
```



In [31]:
```
sns.boxplot(df['FG3_PCT_away'])
plt.title("Box plot of FG3_PCT_away")
plt.show
```

```
C:\Users\swethak\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarn
ing: Pass the following variable as a keyword arg: x. From version 0.12, the only
valid positional argument will be `data`, and passing other arguments without an e
xplicit keyword will result in an error or misinterpretation.
  warnings.warn(
```

Out[31]:
```
<function matplotlib.pyplot.show(close=None, block=None)>
```



In [32]:
```
df['FG3_PCT_away'].value_counts().head()
```

Out[32]:
```
0.333    1470
0.500     932
0.250     853
0.400     817
0.375     595
Name: FG3_PCT_away, dtype: int64
```

In [33]:
```
df['FG_PCT_away'].value_counts()
```

Out[33]:
```
0.500    701
0.494    498
0.506    446
0.488    337
0.432    293
        ...
0.657      1
0.269      1
0.627      1
0.658      1
0.645      1
Name: FG_PCT_away, Length: 390, dtype: int64
```

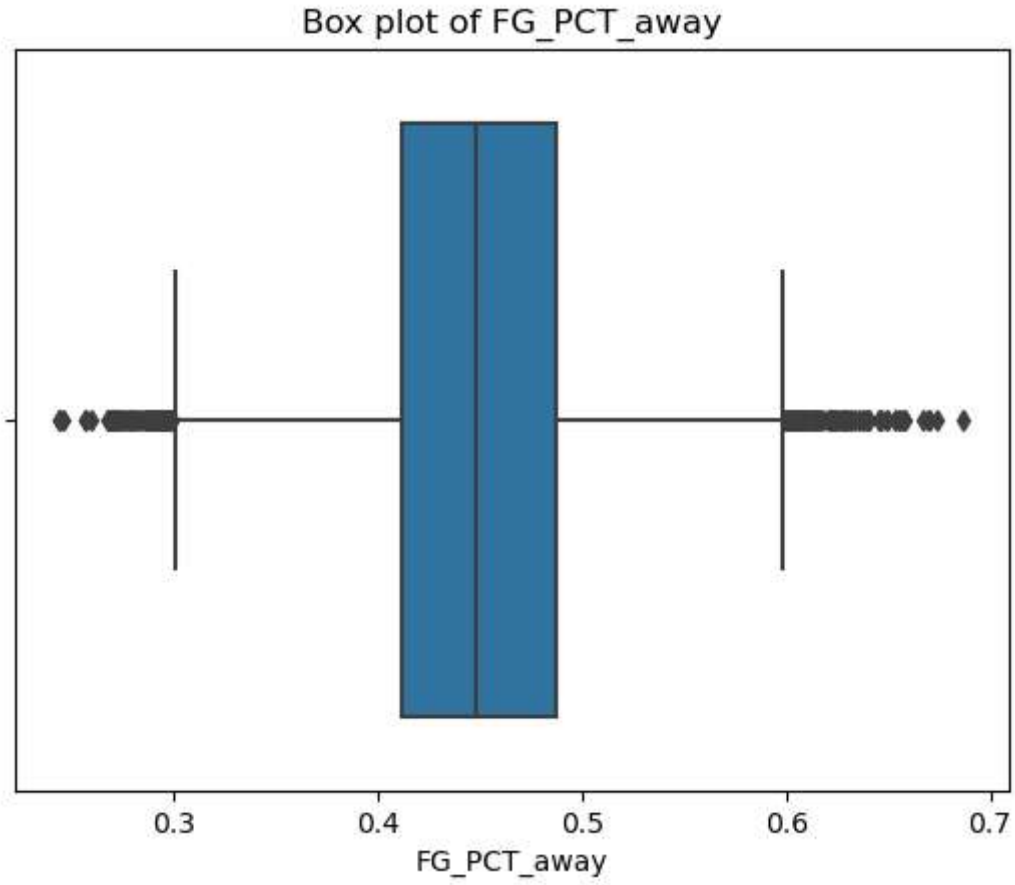
In [34]:
```
df['FG3_PCT_away'].describe()
```

Out[34]:
```
count    25697.000000
mean         0.349413
std          0.110194
min          0.000000
25%          0.278000
50%          0.350000
75%          0.420000
max          1.000000
Name: FG3_PCT_away, dtype: float64
```

In [35]:
```
sns.boxplot(df.FG_PCT_away)
plt.title("Box plot of FG_PCT_away")
plt.show
```

```
C:\Users\swethak\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarn
ing: Pass the following variable as a keyword arg: x. From version 0.12, the only
valid positional argument will be `data`, and passing other arguments without an e
xplicit keyword will result in an error or misinterpretation.
  warnings.warn(
```

Out[35]:
```
<function matplotlib.pyplot.show(close=None, block=None)>
```

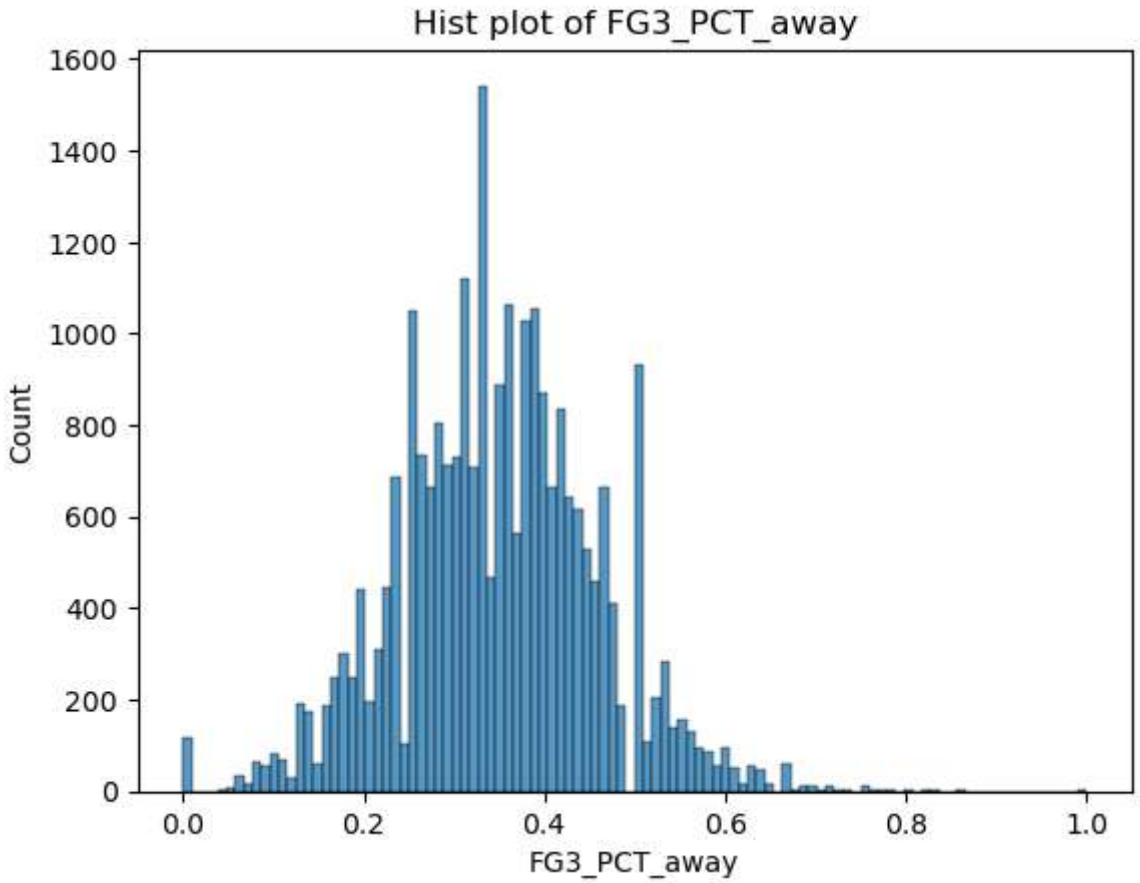


In [36]:
```
max=df['FG3_PCT_away']
max_value=max.max()
min=df['FG3_PCT_away']
min_value=min.min()
print("Max Value",max_value)
print("Min Value",min_value)
```

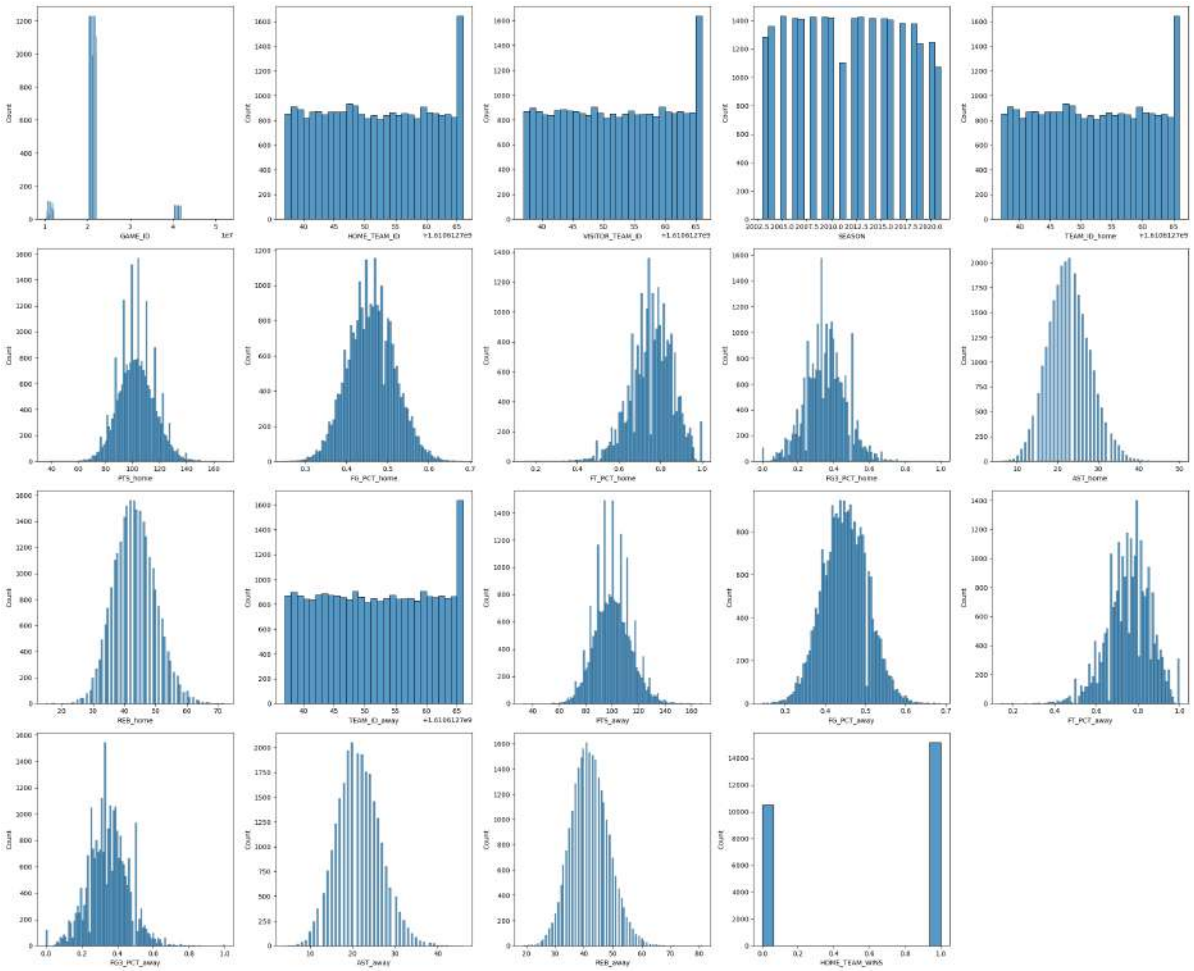
```
Max Value 1.0
Min Value 0.0
```

In [37]:
```
sns.histplot(df['FG3_PCT_away'])
plt.title("Hist plot of FG3_PCT_away")
```

Out[37]:
```
Text(0.5, 1.0, 'Hist plot of FG3_PCT_away')
```

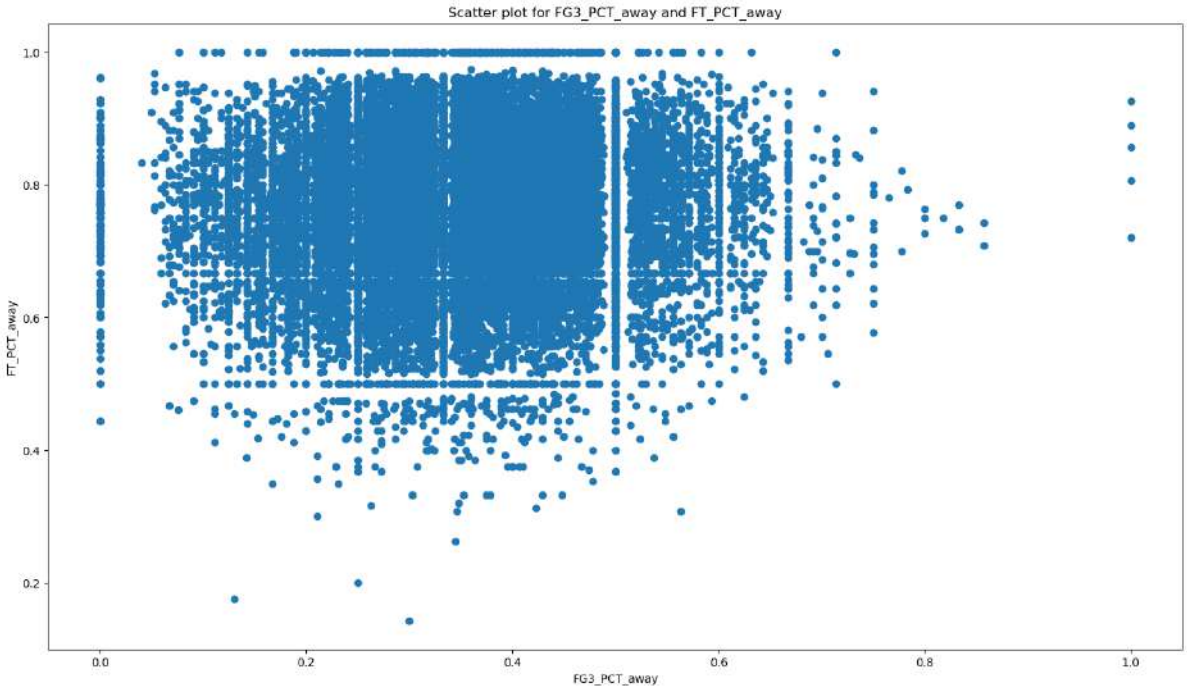


In [38]:
```
#Importing Matplot library in order to visualise the data in barplot

import matplotlib.pyplot as plt

cols = 5
rows = 5
num_cols = df.select_dtypes(exclude='object').columns
fig = plt.figure( figsize=(cols*5, rows*5))
for i, col in enumerate(num_cols):
#for i in num_cols:
    ax=fig.add_subplot(rows,cols,i+1)

    sns.histplot(x = df[col], ax = ax)

fig.tight_layout()
plt.show()
```



In [39]:

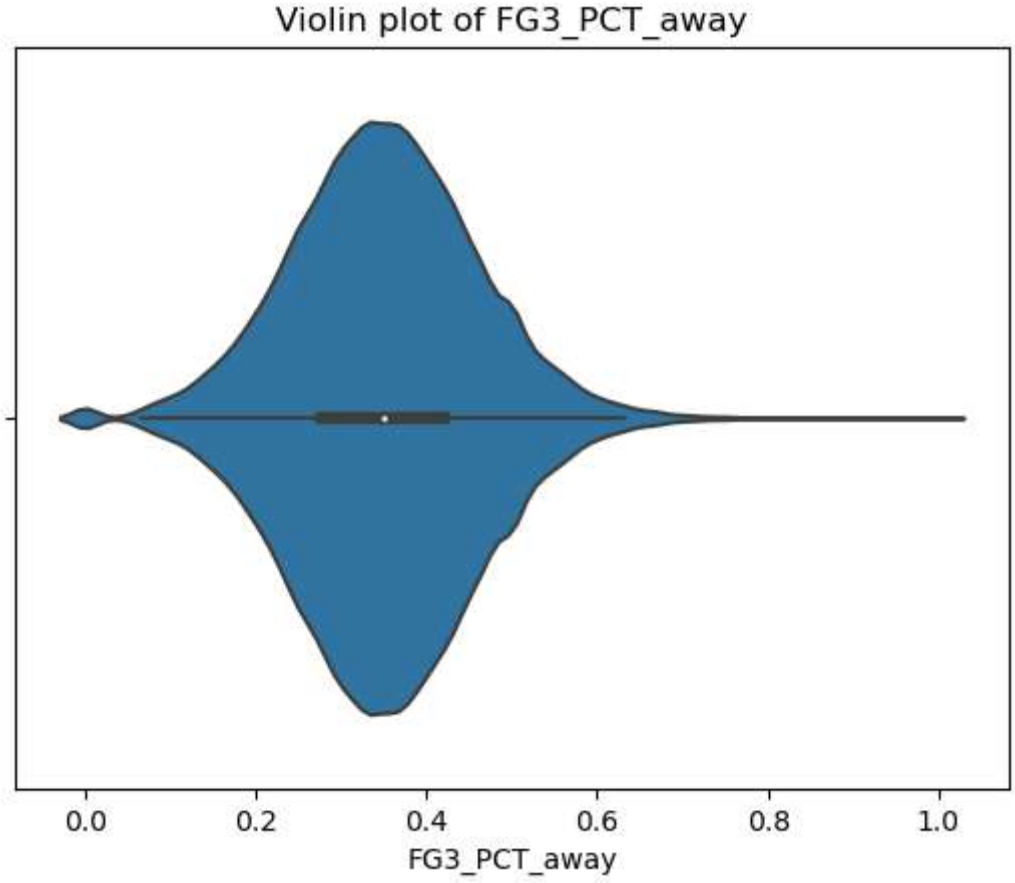
```
import matplotlib.pyplot as plt
fig, ax = plt.subplots(figsize = (18,10))
plt.title("Scatter plot for FG3_PCT_away and FT_PCT_away")
ax.scatter(df['FG3_PCT_away'], df['FT_PCT_away'])

# x-axis label
ax.set_xlabel('FG3_PCT_away')

# y-axis label
ax.set_ylabel('FT_PCT_away')
plt.show()
```



Violin plot

In [40]:
```
sns.violinplot(df["FG3_PCT_away"])
plt.title("Violin plot of FG3_PCT_away")
plt.show()
```

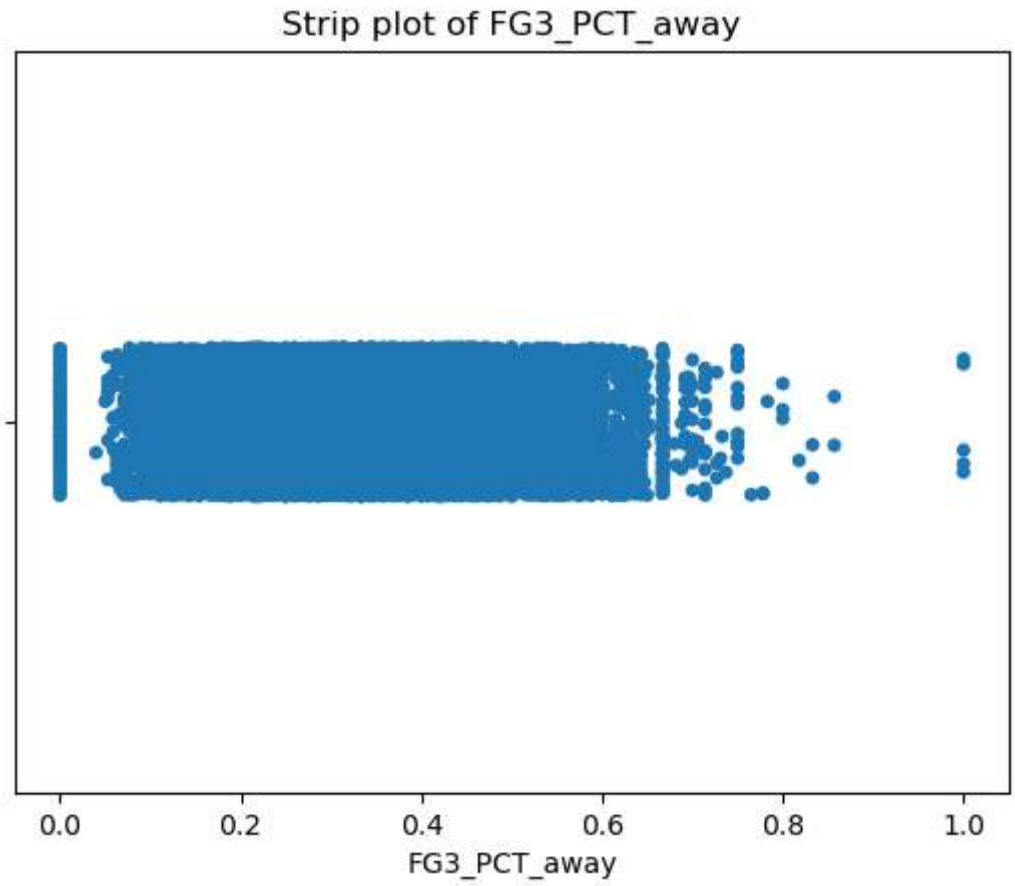
```
C:\Users\swethak\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarn
ing: Pass the following variable as a keyword arg: x. From version 0.12, the only
valid positional argument will be `data`, and passing other arguments without an e
xplicit keyword will result in an error or misinterpretation.
  warnings.warn(
```



Strip plot

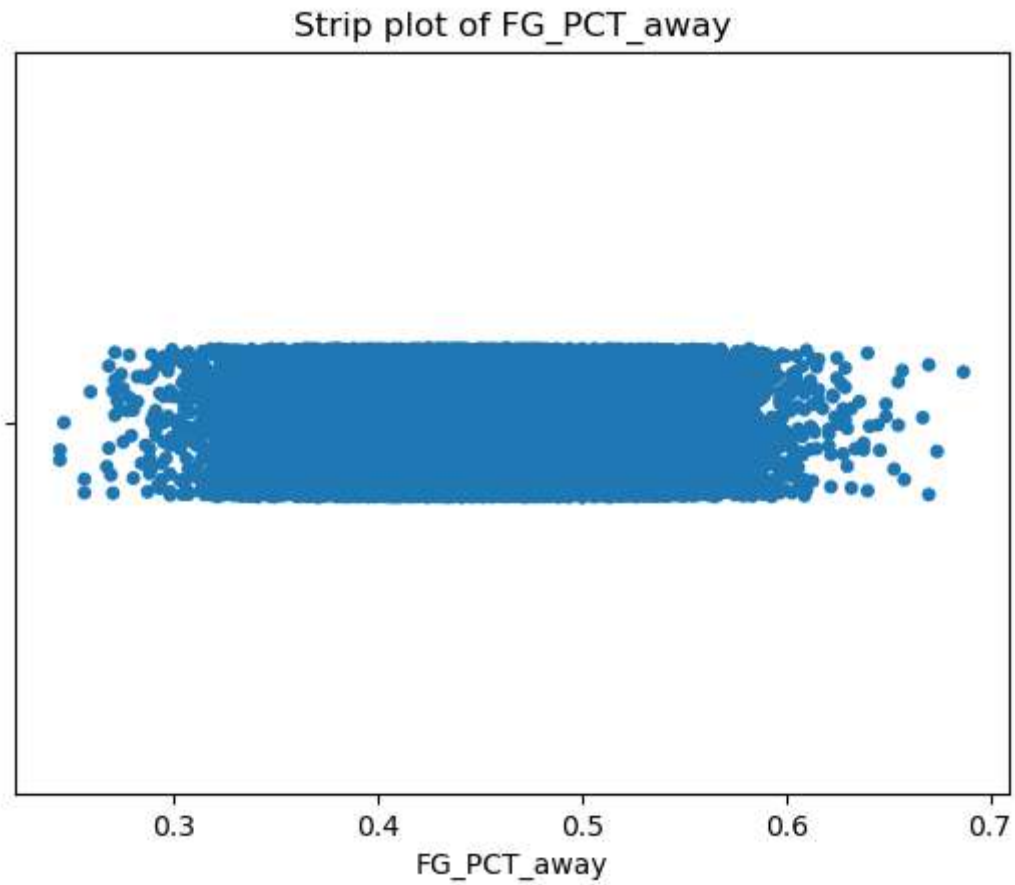
In [41]:
```
sns.stripplot(df["FG3_PCT_away"])
plt.title("Strip plot of FG3_PCT_away")
plt.show()
```

```
C:\Users\swethak\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarn
ing: Pass the following variable as a keyword arg: x. From version 0.12, the only
valid positional argument will be `data`, and passing other arguments without an e
xplicit keyword will result in an error or misinterpretation.
  warnings.warn(
```



In [42]:
```
sns.stripplot(df["FG_PCT_away"])
plt.title("Strip plot of FG_PCT_away")
plt.show()
```

```
C:\Users\swethak\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarn
ing: Pass the following variable as a keyword arg: x. From version 0.12, the only
valid positional argument will be `data`, and passing other arguments without an e
xplicit keyword will result in an error or misinterpretation.
  warnings.warn(
```



Histplot

In [43]:
```
df[['PTS_away','FG_PCT_away','FT_PCT_away','FG3_PCT_away']].describe()
```

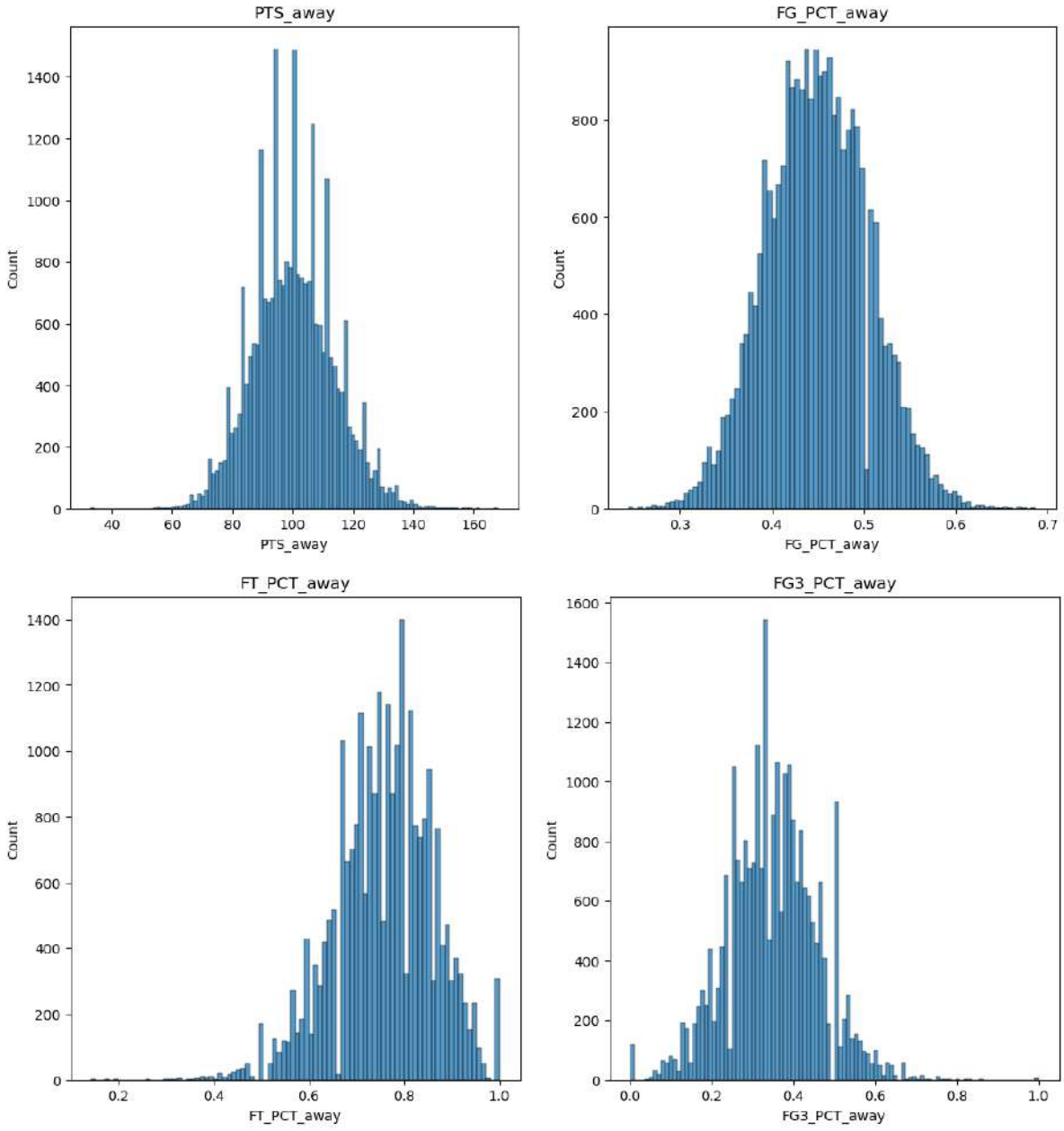
Out[43]:

	PTS_away	FG_PCT_away	FT_PCT_away	FG3_PCT_away
count	25697.000000	25697.000000	25697.000000	25697.000000
mean	100.294120	0.449265	0.758082	0.349413
std	13.343016	0.055528	0.103418	0.110194
min	33.000000	0.244000	0.143000	0.000000
25%	91.000000	0.412000	0.692000	0.278000
50%	100.000000	0.448000	0.765000	0.350000
75%	109.000000	0.487000	0.833000	0.420000
max	168.000000	0.687000	1.000000	1.000000

Ploting the Histplot for the above columns

In [44]:
```
plt.figure(figsize=(20,14))
plt.title("Histplot")
plt.subplot(231)
plt.title("PTS_away")
sns.histplot(df["PTS_away"])
plt.subplot(232)
plt.title("FG_PCT_away")
sns.histplot(df["FG_PCT_away"])
```

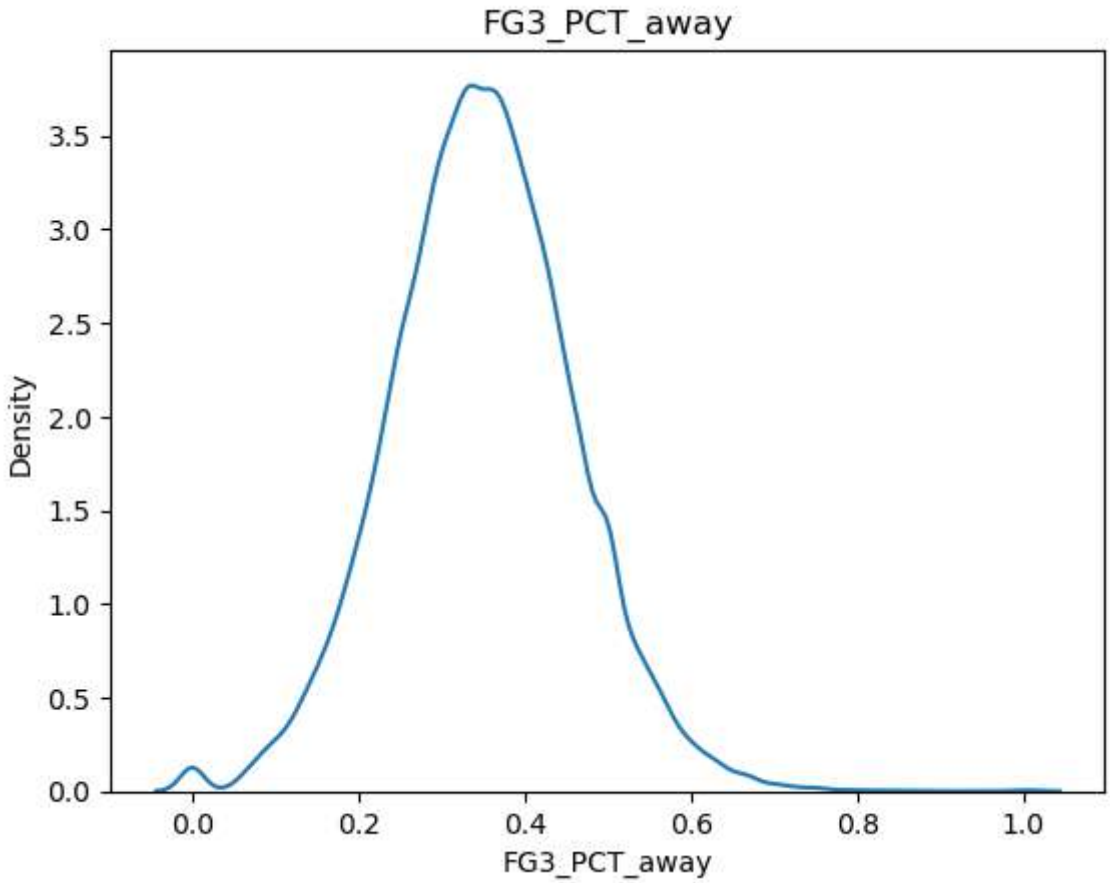
```
plt.figure(figsize=(20,14))
plt.subplot(231)
plt.title("FT_PCT_away")
sns.histplot(df["FT_PCT_away"])
plt.subplot(232)
plt.title("FG3_PCT_away")
sns.histplot(df["FG3_PCT_away"])
plt.show()
```



KDE plot

In [45]:
```
sns.distplot(df['FG3_PCT_away'],hist=False)
plt.title("FG3_PCT_away")
plt.figure(figsize=(20,14))
plt.show()
```

```
C:\Users\swethak\anaconda3\lib\site-packages\seaborn\distributions.py:2619: Future
Warning: `distplot` is a deprecated function and will be removed in a future versi
on. Please adapt your code to use either `displot` (a figure-level function with s
imilar flexibility) or `kdeplot` (an axes-level function for kernel density plot
s).
  warnings.warn(msg, FutureWarning)
```




```
<Figure size 2000x1400 with 0 Axes>
```

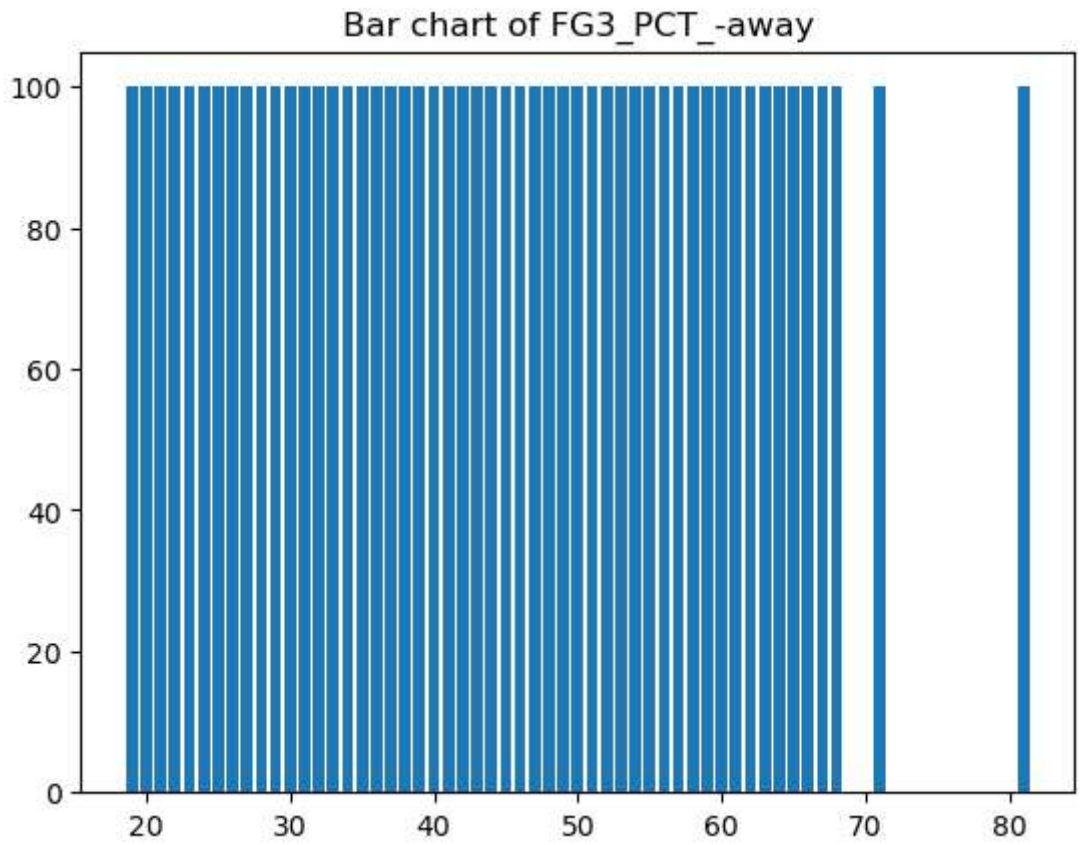
Bar chart

In [46]:
```
df["REB_away"].value_counts()
```

Out[46]:
```
41.0    1607
40.0    1560
42.0    1528
43.0    1505
39.0    1490
44.0    1471
38.0    1407
45.0    1332
37.0    1284
46.0    1230
47.0    1131
36.0    1066
48.0     996
35.0     929
49.0     879
34.0     753
50.0     695
33.0     638
51.0     553
32.0     481
52.0     453
53.0     379
31.0     345
54.0     302
30.0     258
55.0     227
56.0     186
29.0     182
28.0     139
57.0     134
58.0      91
27.0      82
59.0      78
60.0      56
26.0      50
25.0      38
61.0      37
62.0      27
24.0      17
63.0      15
23.0      12
65.0      11
22.0      11
64.0      10
21.0       7
66.0       6
68.0       3
67.0       2
19.0       1
71.0       1
81.0       1
20.0       1
Name: REB_away, dtype: int64
```

Bivariate analysis

In [47]:
```
plt.bar(df["REB_away"],height=100)
plt.title("Bar chart of FG3_PCT_-away")
plt.figure(figsize=(19,90))
plt.show()
```




```
<Figure size 1900x9000 with 0 Axes>
```

In [48]:
```
df["HOME_TEAM_WINS"].value_counts().sort_values()
```

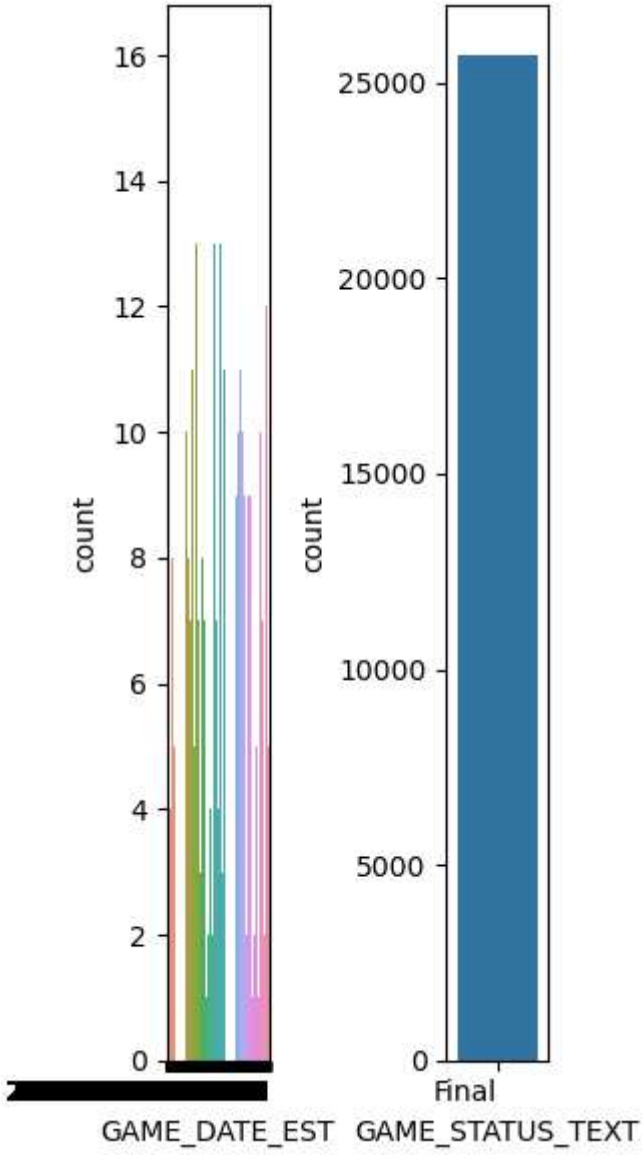
Out[48]:
```
0    10542
1    15155
Name: HOME_TEAM_WINS, dtype: int64
```

In [49]:
```
df.describe(include='object')
```

Out[49]:

	GAME_DATE_EST	GAME_STATUS_TEXT
count	25697	25697
unique	4133	1
top	2020-12-23	Final
freq	16	25697

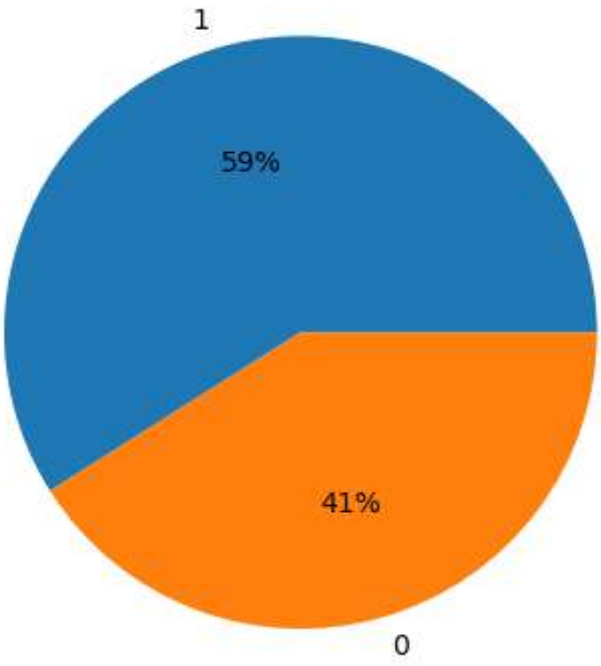
In [50]:
```
import matplotlib.pyplot as plt
cols = 7
rows = 1
fig = plt.figure(figsize= (10,6))
all_categs = df.select_dtypes(include='object')
cat_cols = all_categs.columns[all_categs.nunique() < 5000]
for i, col in enumerate(cat_cols):
    ax=fig.add_subplot(rows, cols, i+1)
    sns.countplot(x=df[col], ax=ax)
    plt.xticks(rotation=0, ha='right')

fig.tight_layout()
plt.show()
```



Pie chart

In [51]:
```
data=df['HOME_TEAM_WINS'].value_counts()
plt.pie(data, labels=data.index, autopct="%.0f%%")
```

Out[51]:
```
([<matplotlib.patches.Wedge at 0x265a74311c0>,
  <matplotlib.patches.Wedge at 0x265a7435430>],
 [Text(-0.3060855872458776, 1.0565564884474214, '1'),
  Text(0.3060855872458774, -1.0565564884474217, '0')],
 [Text(-0.16695577486138774, 0.5763035391531389, '59%'),
  Text(0.16695577486138768, -0.576303539153139, '41%')])
```

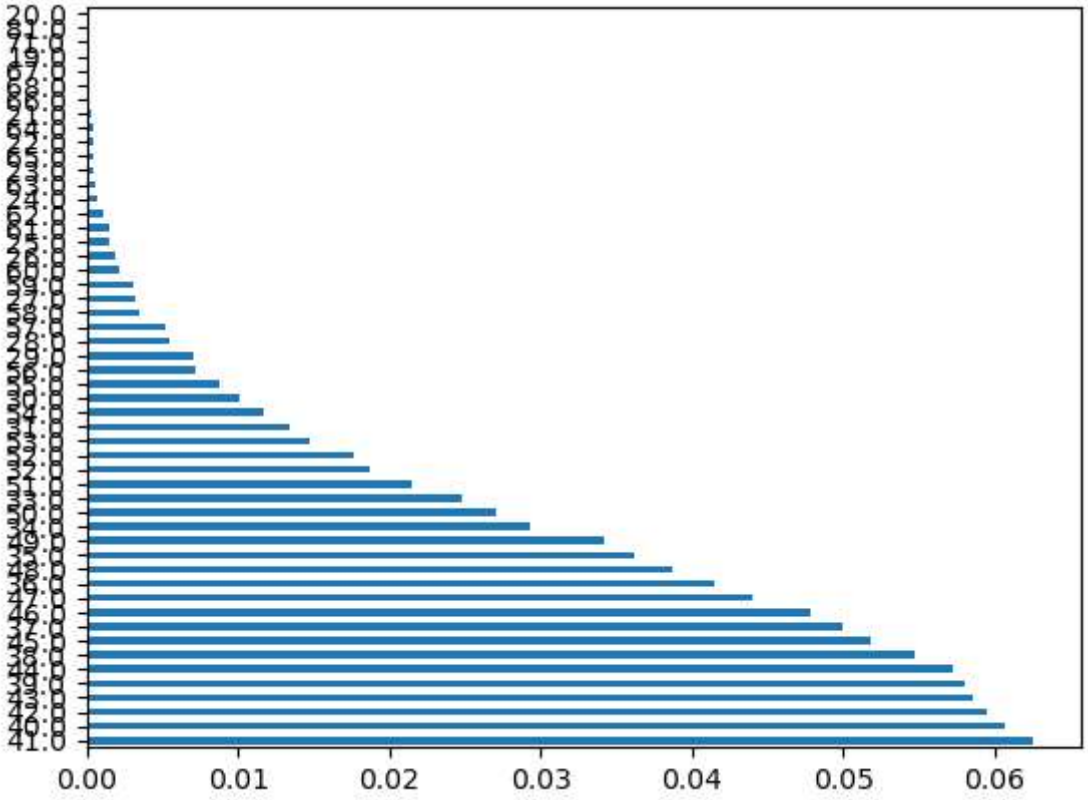


In [52]:
```
df['REB_away'].value_counts(normalize=True)
```

Out[52]:
```
41.0    0.062536
40.0    0.060707
42.0    0.059462
43.0    0.058567
39.0    0.057983
44.0    0.057244
38.0    0.054753
45.0    0.051835
37.0    0.049967
46.0    0.047866
47.0    0.044013
36.0    0.041483
48.0    0.038759
35.0    0.036152
49.0    0.034206
34.0    0.029303
50.0    0.027046
33.0    0.024828
51.0    0.021520
32.0    0.018718
52.0    0.017629
53.0    0.014749
31.0    0.013426
54.0    0.011752
30.0    0.010040
55.0    0.008834
56.0    0.007238
29.0    0.007083
28.0    0.005409
57.0    0.005215
58.0    0.003541
27.0    0.003191
59.0    0.003035
60.0    0.002179
26.0    0.001946
25.0    0.001479
61.0    0.001440
62.0    0.001051
24.0    0.000662
63.0    0.000584
23.0    0.000467
65.0    0.000428
22.0    0.000428
64.0    0.000389
21.0    0.000272
66.0    0.000233
68.0    0.000117
67.0    0.000078
19.0    0.000039
71.0    0.000039
81.0    0.000039
20.0    0.000039
Name: REB_away, dtype: float64
```

In [53]:
```
df['REB_away'].value_counts(normalize=True).plot.barh()
```

Out[53]:
```
<AxesSubplot:>
```

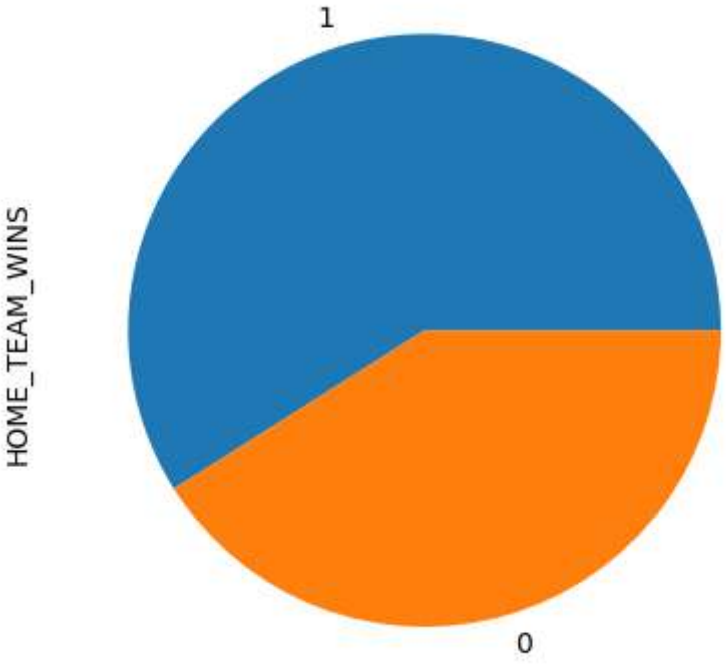


In [54]:
```
df['HOME_TEAM_WINS'].value_counts()
```

Out[54]:
```
1    15155
0    10542
Name: HOME_TEAM_WINS, dtype: int64
```

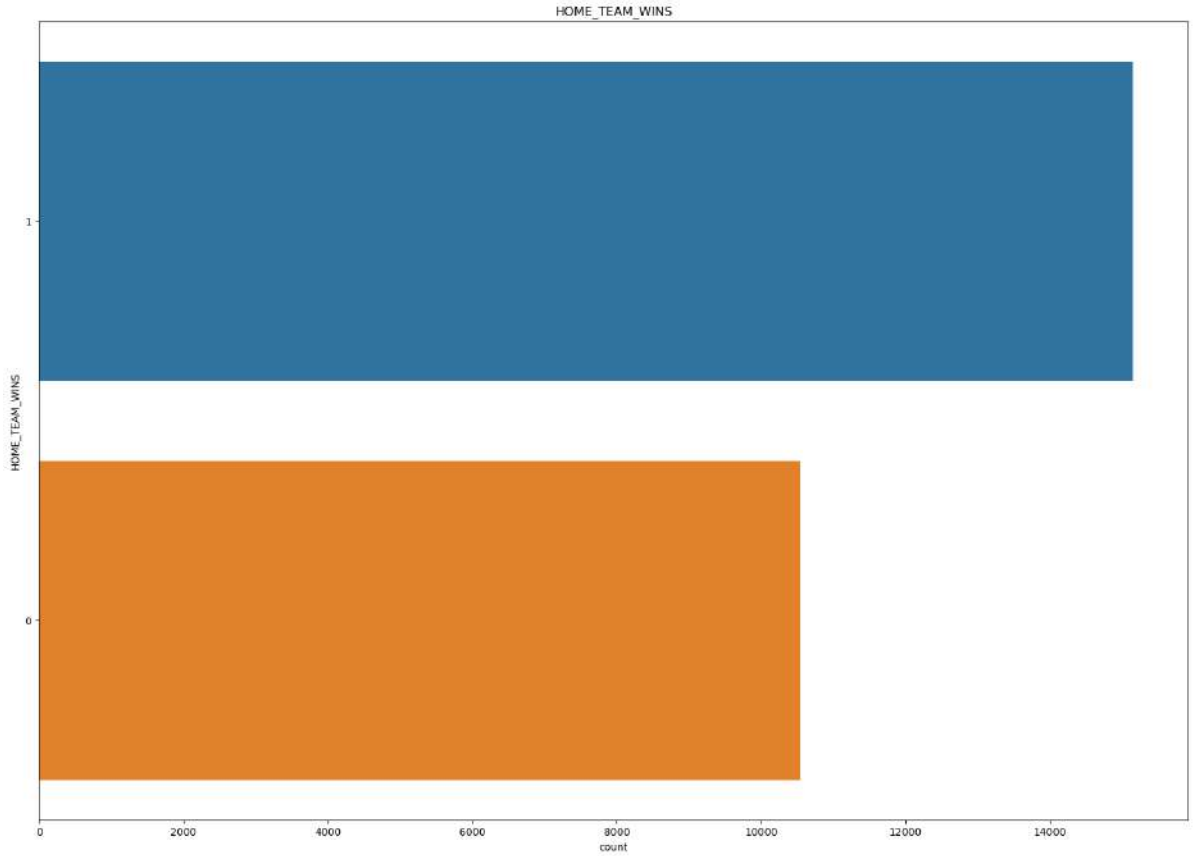
In [55]:
```
df['HOME_TEAM_WINS'].value_counts().plot.pie()
```

Out[55]:
```
<AxesSubplot:ylabel='HOME_TEAM_WINS'>
```

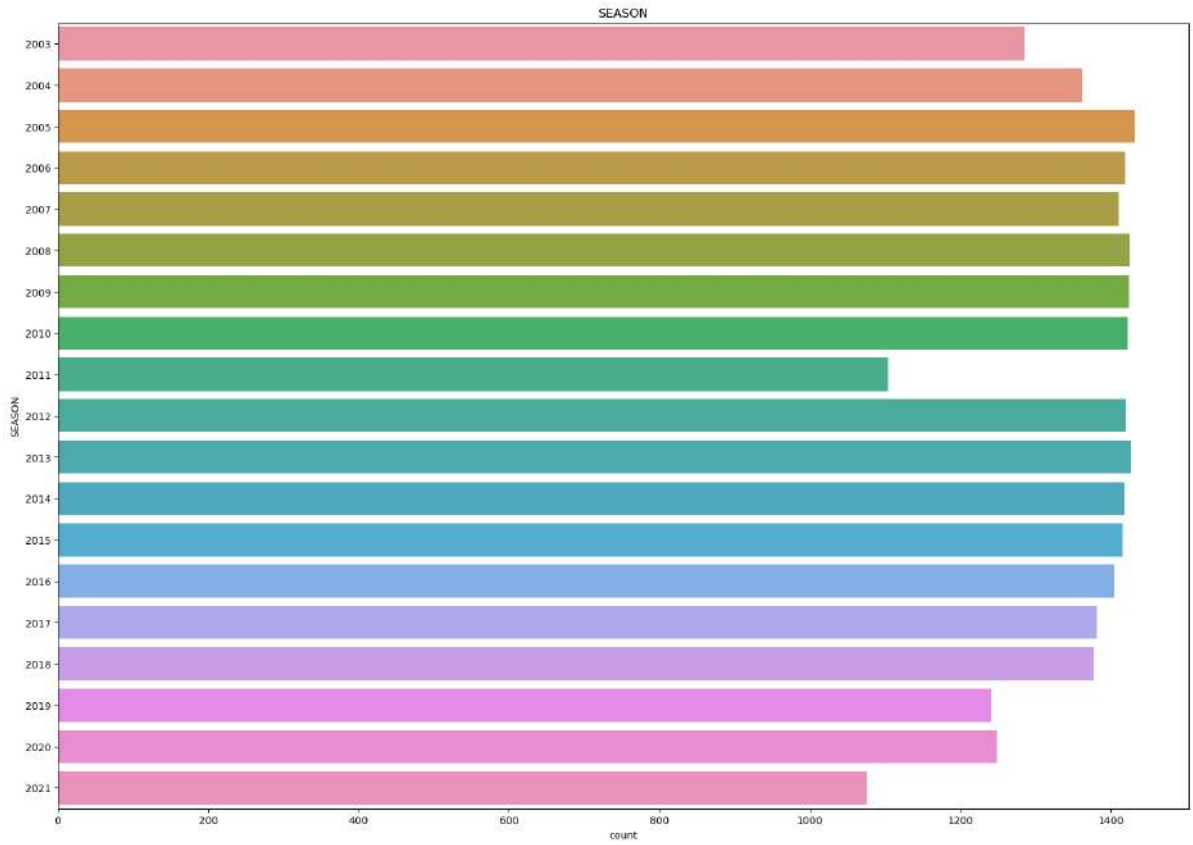


In [56]:
```
plt.figure(figsize=(20,14))
plt.title("HOME_TEAM_WINS")
```

```
sns.countplot(y="HOME_TEAM_WINS",data=df,order=df["HOME_TEAM_WINS"].value_counts()
plt.show()
```

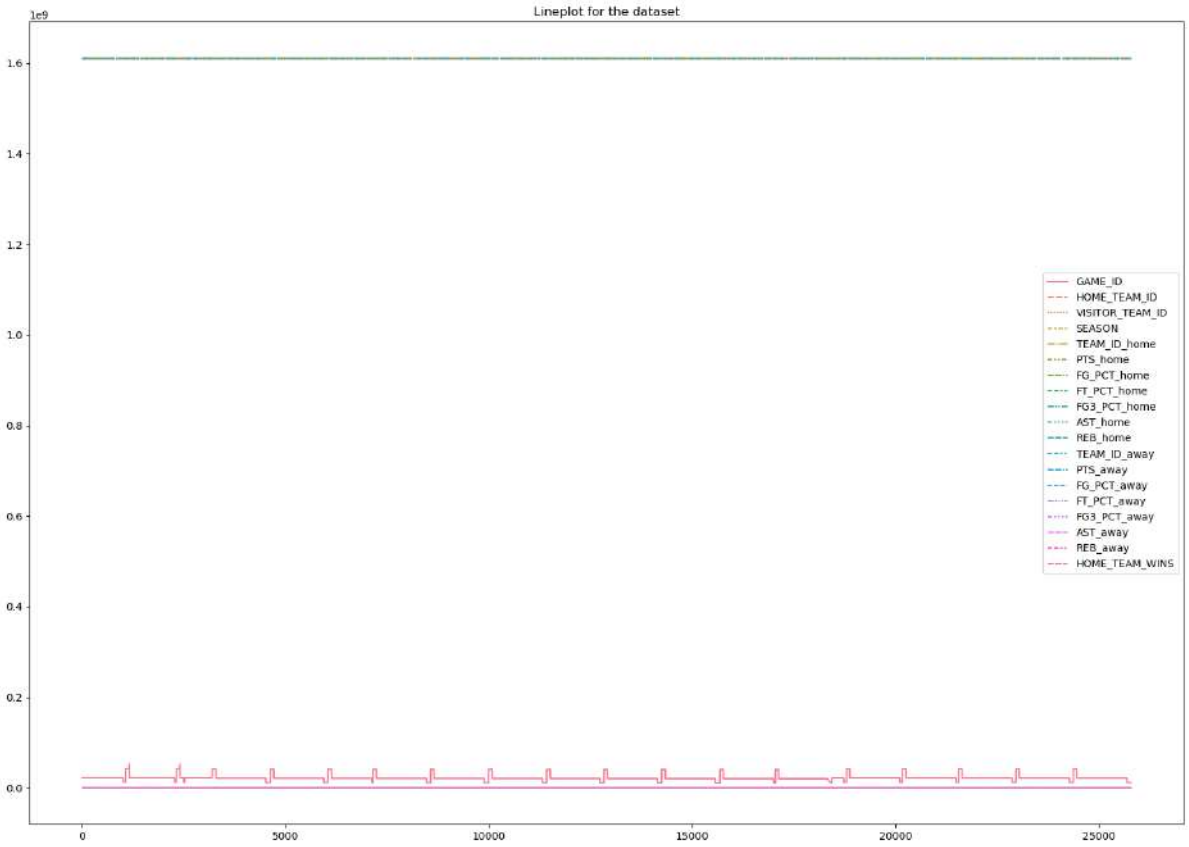


In [57]:
```
plt.figure(figsize=(20,14))
plt.title("SEASON")
sns.countplot(y="SEASON",data=df,order=df["SEASON"].value_counts().index.sort_valu
plt.show()
```

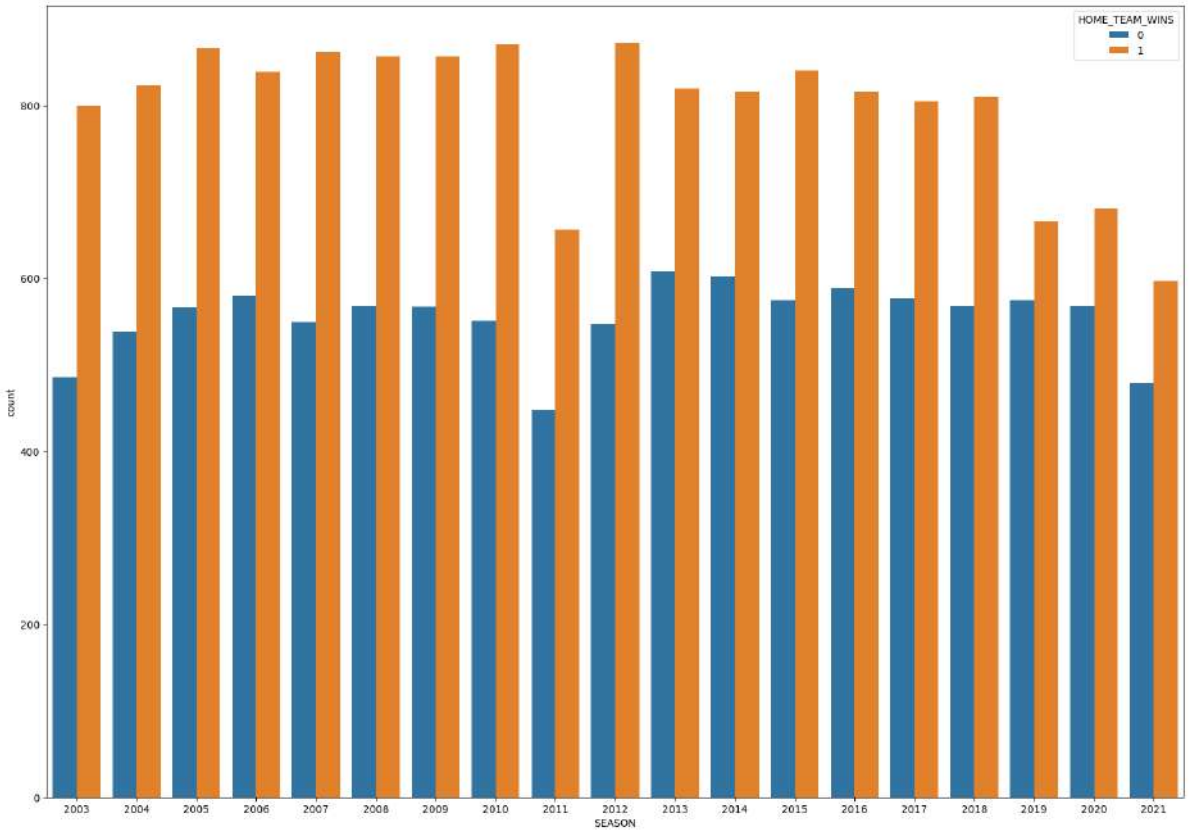


In [58]:
```
plt.figure(figsize=(20,14))
plt.title("Lineplot for the dataset")
```

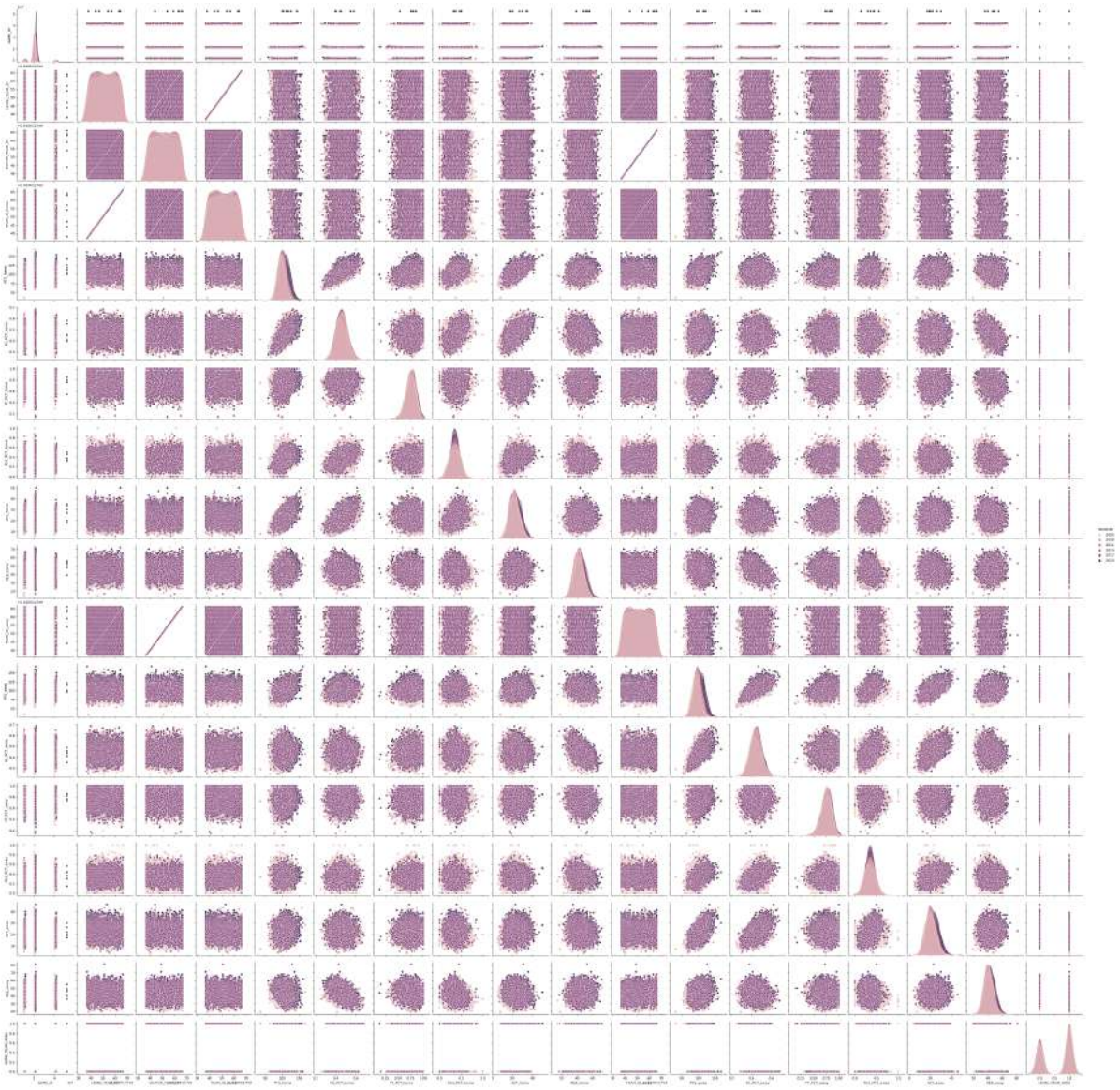
```
sns.lineplot(data=df)
plt.show()
```



In [59]:
```
plt.figure(figsize=(20,14))
sns.countplot(x="SEASON",hue="HOME_TEAM_WINS",data=df,order=df["SEASON"].value_cou
plt.show()
```

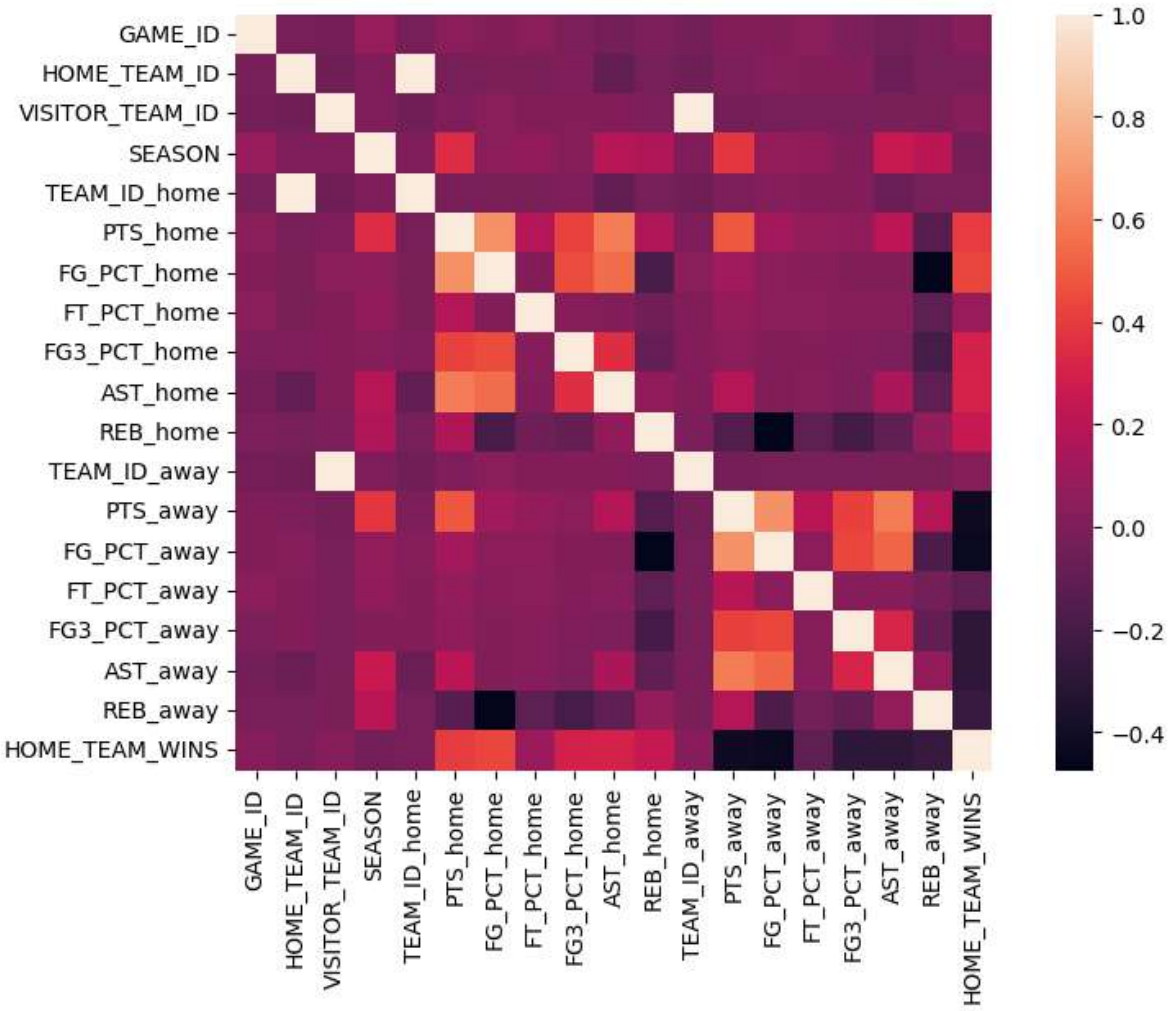


In [60]:
```
# Shows the pairplot with resppect to Year_of_Release
sns.pairplot(df,hue="SEASON")
plt.show()
```



Heatmap graph

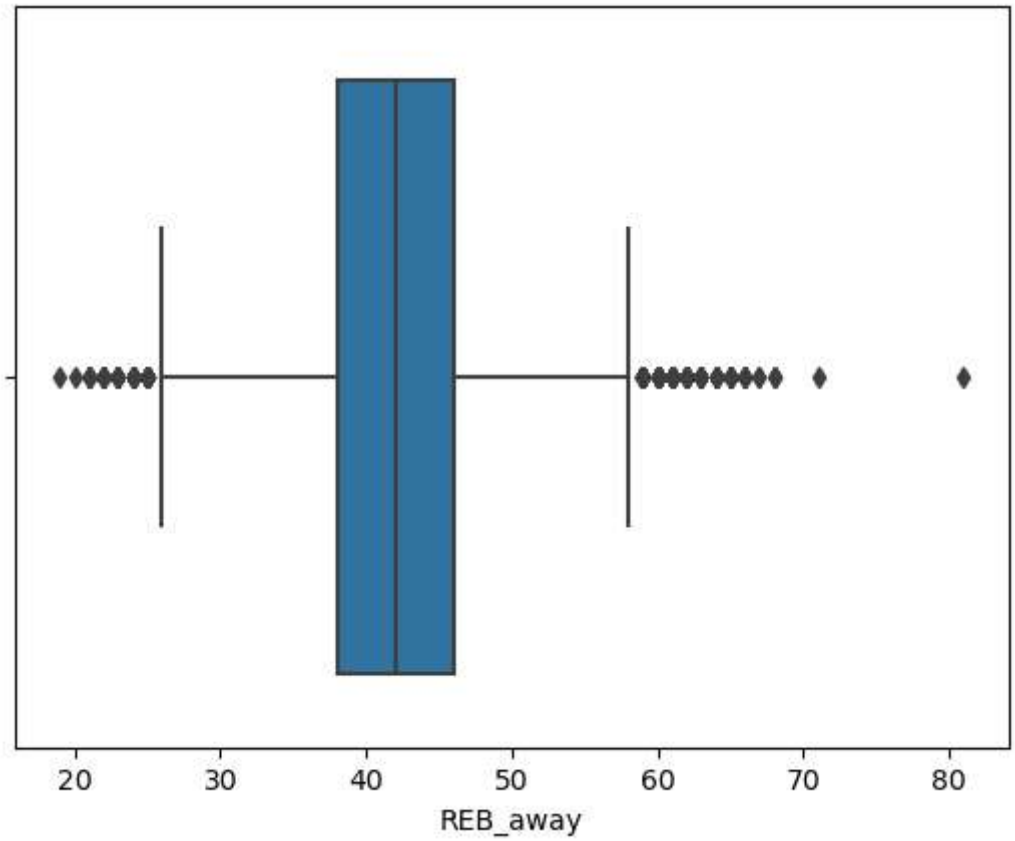
In [61]:
```
# Shows the heatmap graph
corr_df = df.corr()
plt.figure(figsize=(10,6))
sns.heatmap(corr_df,square=True)
plt.show()
```



Outliers

In [62]:
```
sns.boxplot(df["REB_away"])
plt.show()
```

```
C:\Users\swethak\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarn
ing: Pass the following variable as a keyword arg: x. From version 0.12, the only
valid positional argument will be `data`, and passing other arguments without an e
xplicit keyword will result in an error or misinterpretation.
  warnings.warn(
```



In [63]:
```
#outliers are present in the column REB_away
print(np.where(df['REB_away']>0))
```

```
(array([    0,     1,     2, ..., 25694, 25695, 25696], dtype=int64),)
```

Outliers using z-score method

In [64]:
```
from scipy import stats
```

In [65]:
```
z=stats.zscore(df["REB_away"])
z
```

Out[65]:
```
0        0.599810
1       -0.319473
2        0.753024
3       -0.472687
4        1.212665
           ...
25791    0.140168
25792    0.753024
25793    0.140168
25794    0.446596
25795   -0.013046
Name: REB_away, Length: 25697, dtype: float64
```

In [66]:
```
#position of z-score
as1=np.where(z>0)
as1
```

Out[66]:
```
(array([    0,     2,     4, ..., 25693, 25694, 25695], dtype=int64),)
```

Detection of outliers using IQR method

In [67]:
```
Q1=np.percentile(df["REB_away"],25,interpolation='midpoint')
Q3=np.percentile(df["REB_away"],75,interpolation='midpoint')
```

```
IQR=Q3-Q1
print(Q1)
print(Q3)
print(IQR)
print("shape",df.shape)
```

```
38.0
46.0
8.0
shape (25697, 21)
```

In [68]:

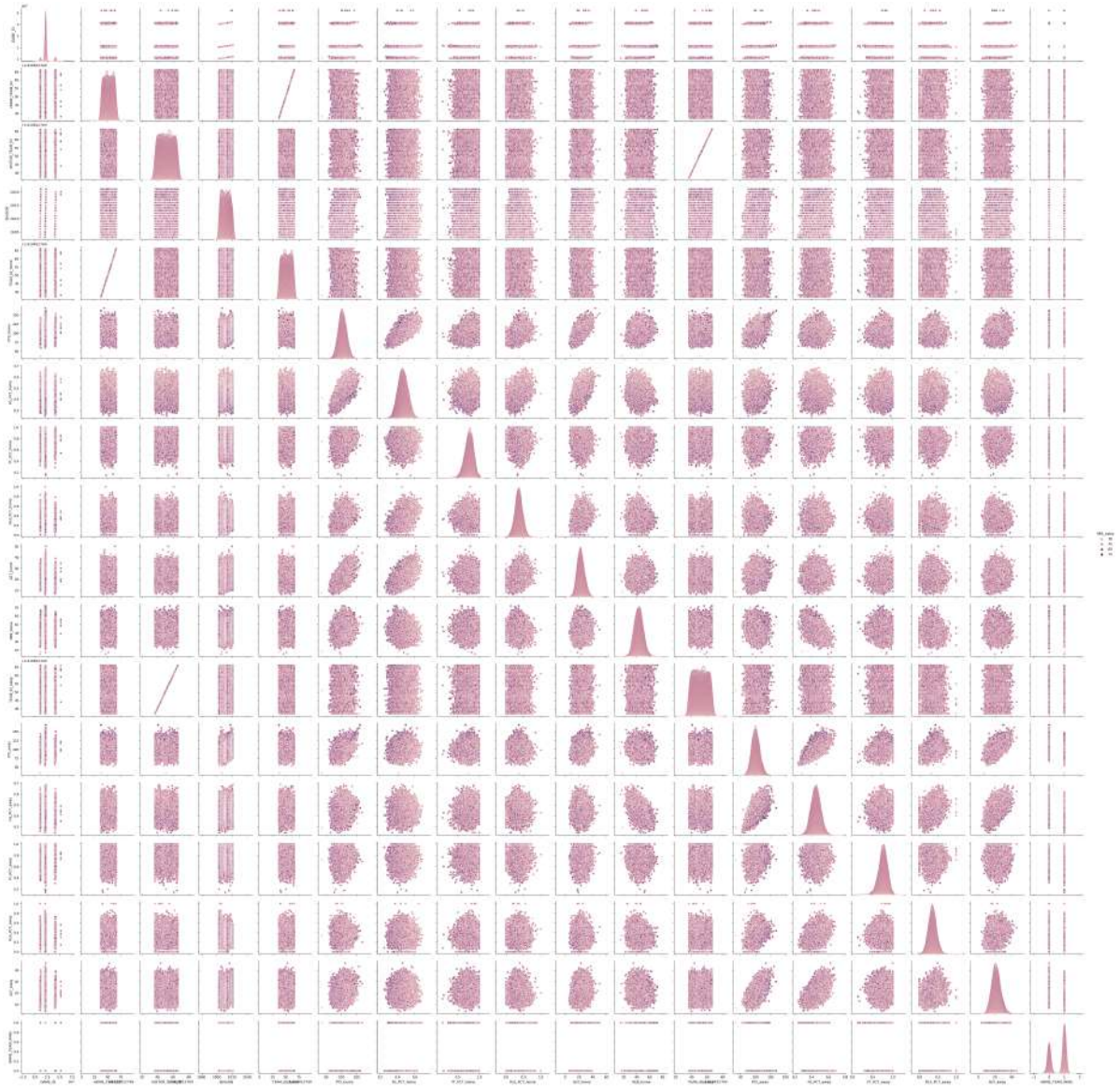
```
upper=df["REB_away"]>=(Q3+1.5*IQR)
print("upper: ",upper)
print(np.where(upper))
lower=df["REB_away"]<=(Q3-1.5*IQR)
print("lower: ",lower)
print(np.where(lower))
print("new shape",df.shape)
```

```
upper:  0        False
1        False
2        False
3        False
4        False
         ...
25791    False
25792    False
25793    False
25794    False
25795    False
Name: REB_away, Length: 25697, dtype: bool
(array([  100,   108,   123,   147,   154,   179,   215,   217,   261,
         281,   462,   558,   752,   877,   882,   883,   943,   948,
         967,   980,   987,  1016,  1023,  1052,  1088,  1118,  1236,
        1352,  1361,  1366,  1377,  1627,  1636,  1694,  1731,  1918,
        1941,  1971,  1981,  2069,  2100,  2178,  2188,  2195,  2252,
        2295,  2315,  2372,  2467,  2482,  2581,  2587,  2597,  2601,
        2618,  2672,  2713,  2717,  2763,  2862,  2895,  2904,  2924,
        2943,  3007,  3008,  3023,  3037,  3040,  3114,  3115,  3126,
        3145,  3200,  3224,  3269,  3296,  3348,  3366,  3378,  3393,
        3541,  3583,  3610,  3712,  3725,  3855,  3946,  3982,  4019,
        4025,  4047,  4185,  4348,  4486,  4533,  4536,  4646,  4723,
        4737,  4807,  5008,  5050,  5186,  5377,  5515,  5526,  5546,
        5567,  5759,  5780,  5796,  5808,  5864,  5922,  5970,  6264,
        6283,  6352,  6401,  6410,  6767,  6889,  7066,  7107,  7371,
        7860,  7885,  7939,  8187,  8293,  8491,  8673,  8832,  8998,
        9054,  9126,  9200,  9216,  9327,  9637,  9745,  9761,  9922,
        9933,  9988, 10113, 10162, 10703, 10748, 10772, 11172, 11181,
       11190, 11295, 11465, 11550, 11551, 11862, 12119, 12536, 12636,
       12776, 12804, 12976, 13248, 13375, 13447, 13468, 14488, 14617,
       15238, 15267, 15268, 15399, 15465, 15615, 15858, 15925, 16079,
       16206, 16773, 16830, 16848, 16853, 17102, 17225, 17368, 17542,
       17546, 17977, 18033, 18062, 18194, 18256, 18313, 18315, 18437,
       18559, 18588, 18603, 18643, 18645, 18651, 18654, 18668, 18691,
       18694, 18708, 18729, 18795, 18797, 18859, 18869, 18894, 18907,
       18919, 18926, 19029, 19035, 19076, 19081, 19112, 19136, 19137,
       19140, 19147, 19161, 19217, 19308, 19345, 19367, 19456, 19489,
       19520, 19555, 19590, 19649, 19758, 19836, 19941, 20035, 20044,
       20045, 20048, 20129, 20255, 20300, 20313, 20319, 20483, 20512,
       20556, 20557, 20583, 20644, 20755, 20786, 20799, 20947, 20963,
       21142, 21285, 21374, 21462, 21474, 21584, 21680, 21681, 21871,
       21900, 21991, 22029, 22050, 22288, 22320, 22440, 22460, 22495,
       22563, 22613, 22627, 22694, 22723, 22760, 22840, 22852, 22994,
       23069, 23153, 23161, 23187, 23201, 23211, 23282, 23357, 23466,
       23500, 23549, 23632, 23753, 23756, 23787, 23839, 24062, 24075,
       24086, 24148, 24176, 24256, 24275, 24289, 24298, 24356, 24375,
       24388, 24507, 24562, 24598, 24631, 24691, 24695, 24727, 24747,
       24757, 24772, 24832, 24855, 24916, 24970, 25063, 25080, 25155,
       25205, 25218, 25321, 25517, 25647], dtype=int64),)
lower:  0        False
1        False
2        False
3        False
4        False
         ...
25791    False
25792    False
25793    False
25794    False
25795    False
Name: REB_away, Length: 25697, dtype: bool
(array([    5,    20,    39, ..., 25654, 25659, 25689], dtype=int64),)
new shape (25697, 21)
```

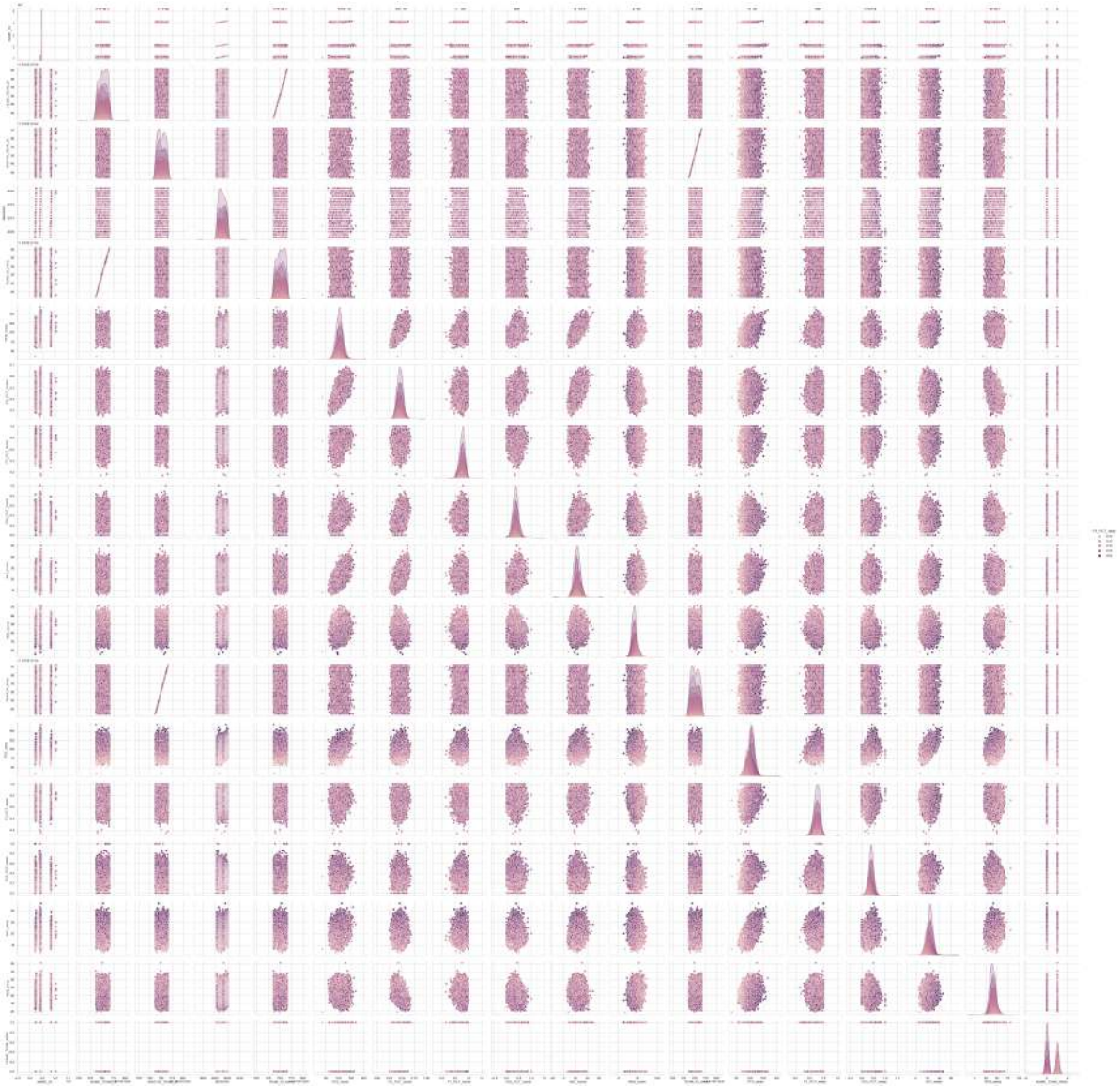
Multivarient Analysis

Pairplot

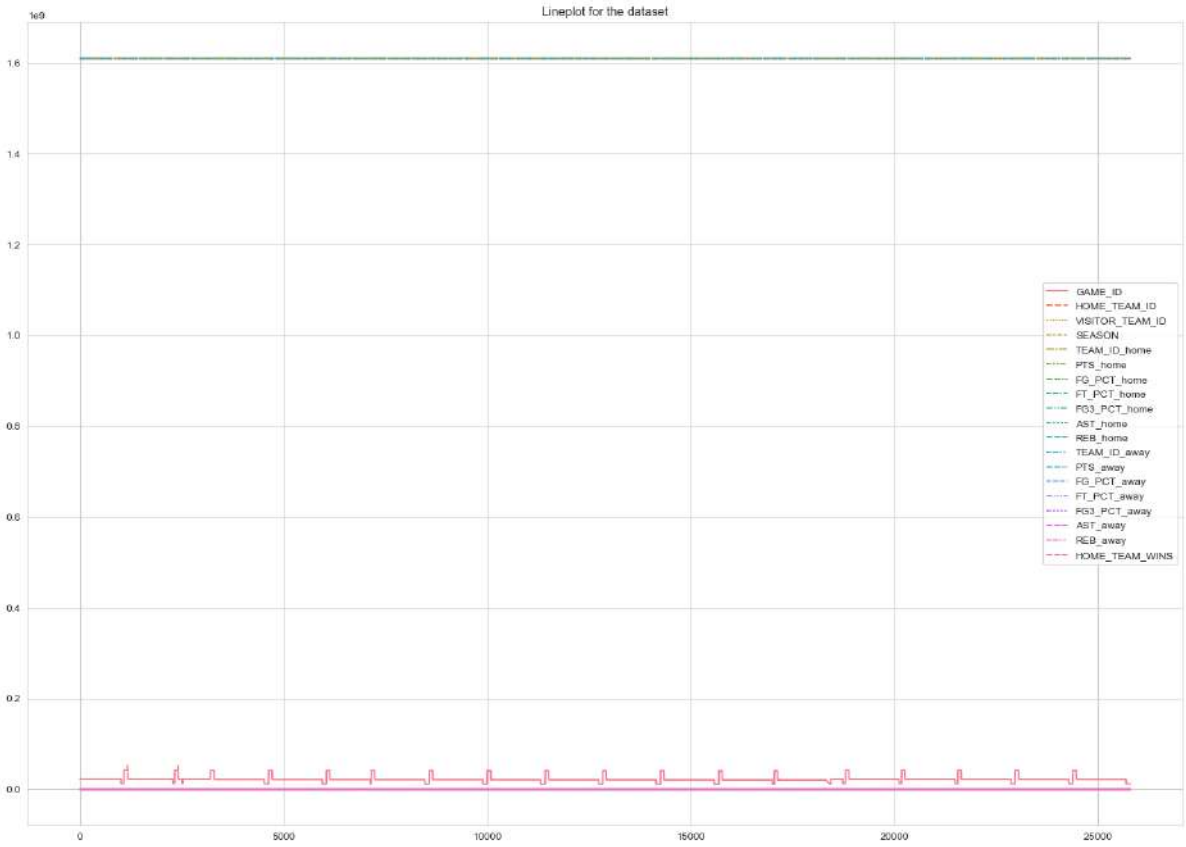
In [69]:
```
#shows thr pairplot with resppect to REB_away
plt.show()
sns.pairplot(df,hue="REB_away")
plt.show()
```



In [70]:
```
#shows thr pairplot with resppect to FG_PCT_away
plt.close()
sns.set_style("whitegrid")
sns.pairplot(df,hue="FG_PCT_away")
plt.show()
```

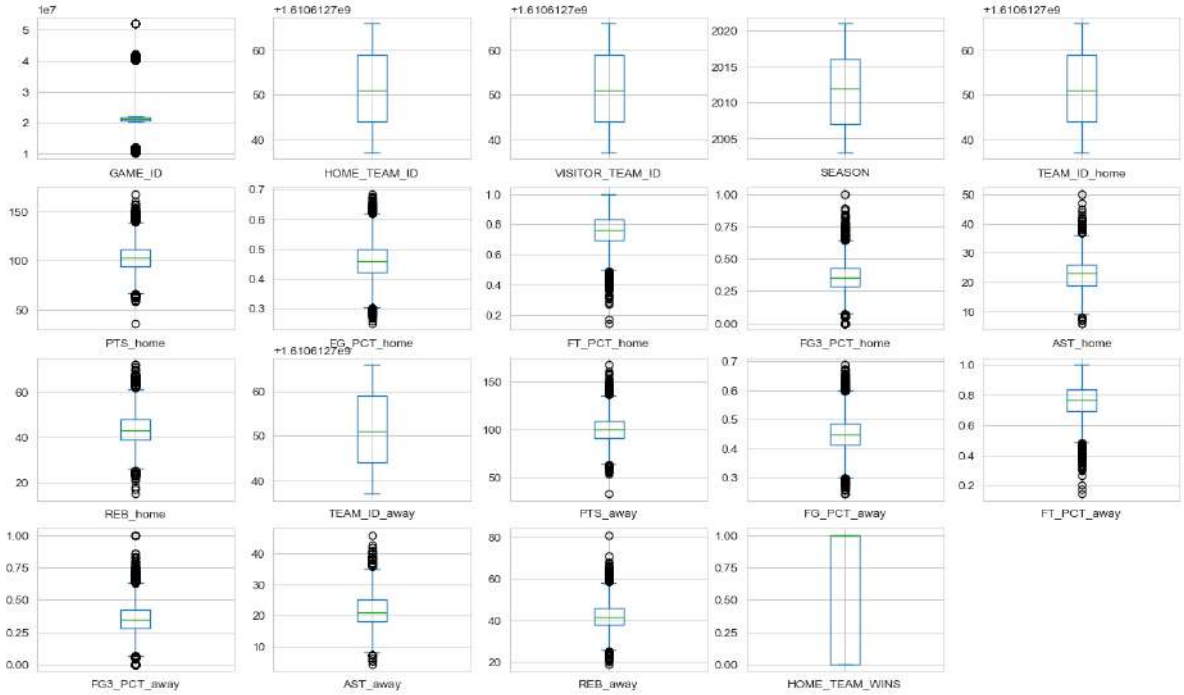


In [71]:
```
#shows the lineplot with resppect to dataset
plt.figure(figsize=(20,14))
plt.title("Lineplot for the dataset")
sns.lineplot(data=df)
plt.show()
```



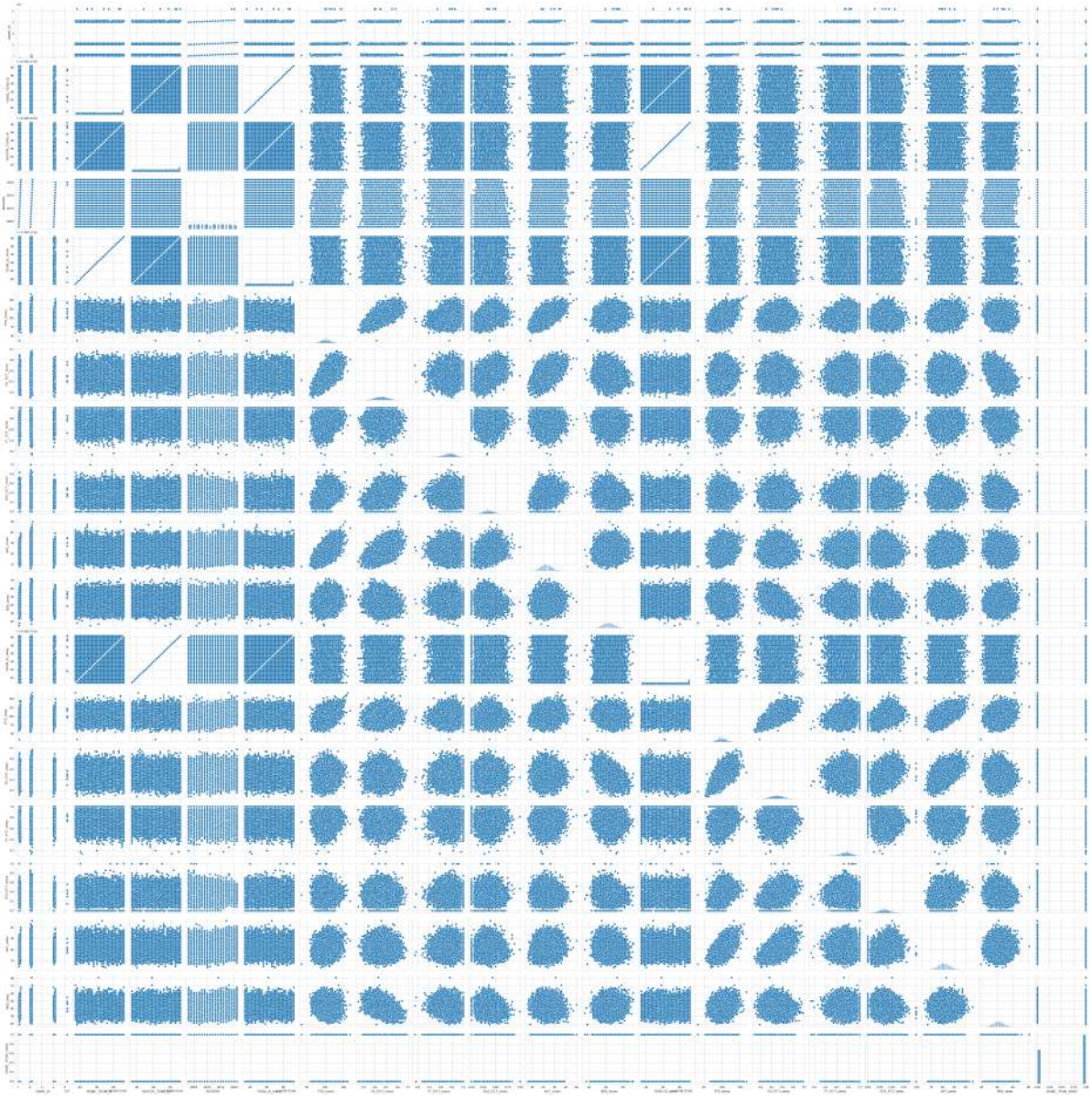
In [72]:
```
df.plot(kind='box', subplots=True, layout=(8,5), figsize=(17,20))
```

Out[72]:
```
GAME_ID              AxesSubplot(0.125,0.798085;0.133621x0.0819149)
HOME_TEAM_ID       AxesSubplot(0.285345,0.798085;0.133621x0.0819149)
VISITOR_TEAM_ID     AxesSubplot(0.44569,0.798085;0.133621x0.0819149)
SEASON             AxesSubplot(0.606034,0.798085;0.133621x0.0819149)
TEAM_ID_home       AxesSubplot(0.766379,0.798085;0.133621x0.0819149)
PTS_home              AxesSubplot(0.125,0.699787;0.133621x0.0819149)
FG_PCT_home        AxesSubplot(0.285345,0.699787;0.133621x0.0819149)
FT_PCT_home         AxesSubplot(0.44569,0.699787;0.133621x0.0819149)
FG3_PCT_home       AxesSubplot(0.606034,0.699787;0.133621x0.0819149)
AST_home           AxesSubplot(0.766379,0.699787;0.133621x0.0819149)
REB_home              AxesSubplot(0.125,0.601489;0.133621x0.0819149)
TEAM_ID_away       AxesSubplot(0.285345,0.601489;0.133621x0.0819149)
PTS_away            AxesSubplot(0.44569,0.601489;0.133621x0.0819149)
FG_PCT_away        AxesSubplot(0.606034,0.601489;0.133621x0.0819149)
FT_PCT_away        AxesSubplot(0.766379,0.601489;0.133621x0.0819149)
FG3_PCT_away          AxesSubplot(0.125,0.503191;0.133621x0.0819149)
AST_away           AxesSubplot(0.285345,0.503191;0.133621x0.0819149)
REB_away            AxesSubplot(0.44569,0.503191;0.133621x0.0819149)
HOME_TEAM_WINS     AxesSubplot(0.606034,0.503191;0.133621x0.0819149)
dtype: object
```



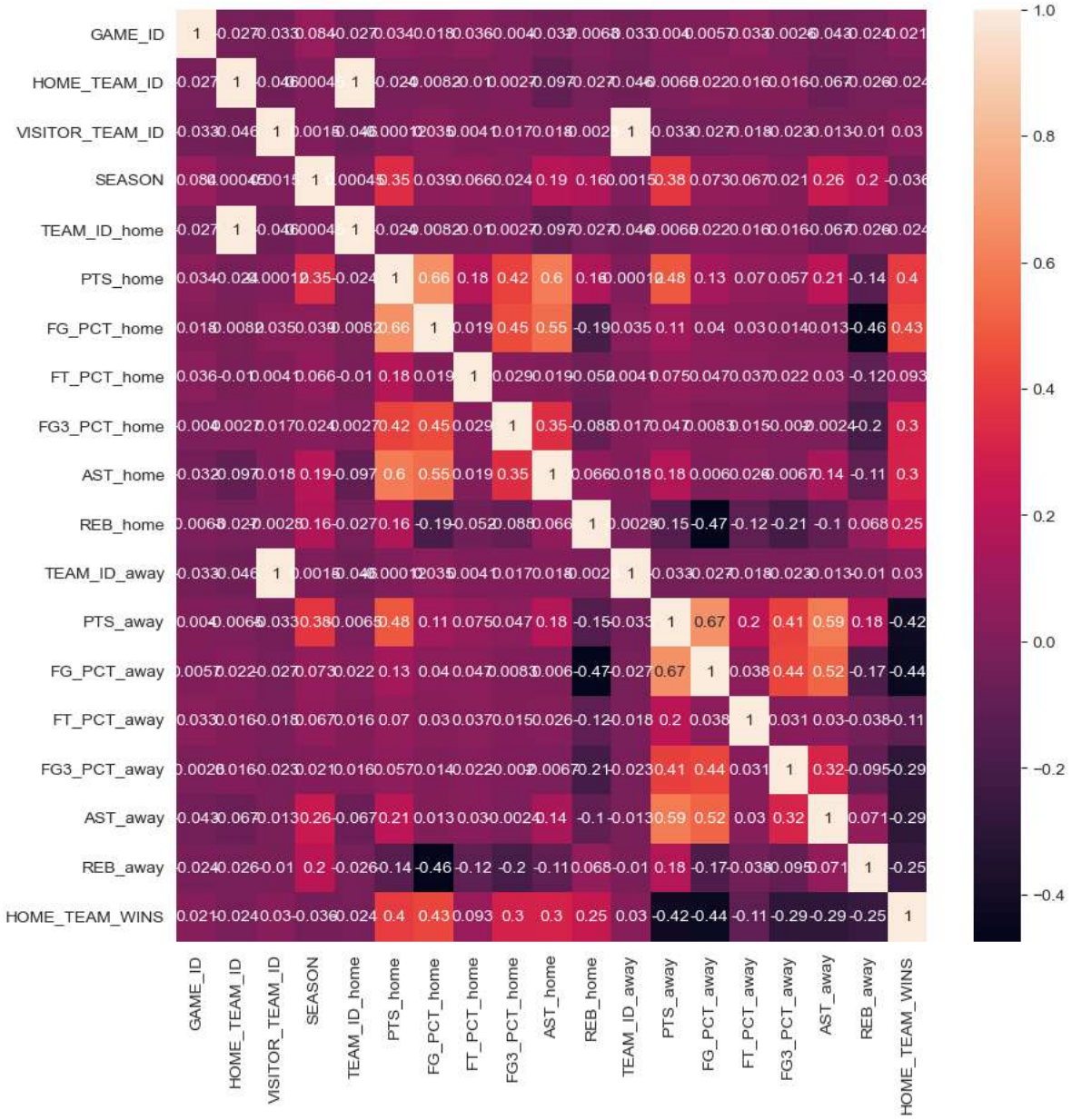
In [73]:
```
sns.pairplot(df)
plt.show
```

Out[73]:
```
<function matplotlib.pyplot.show(close=None, block=None)>
```



In [74]:
```
## Co-relation matrix
fig,ax = plt.subplots(figsize = (10,10))
corr =df.corr()
sns.heatmap(corr,annot=True)
```

Out[74]:
```
<AxesSubplot:>
```

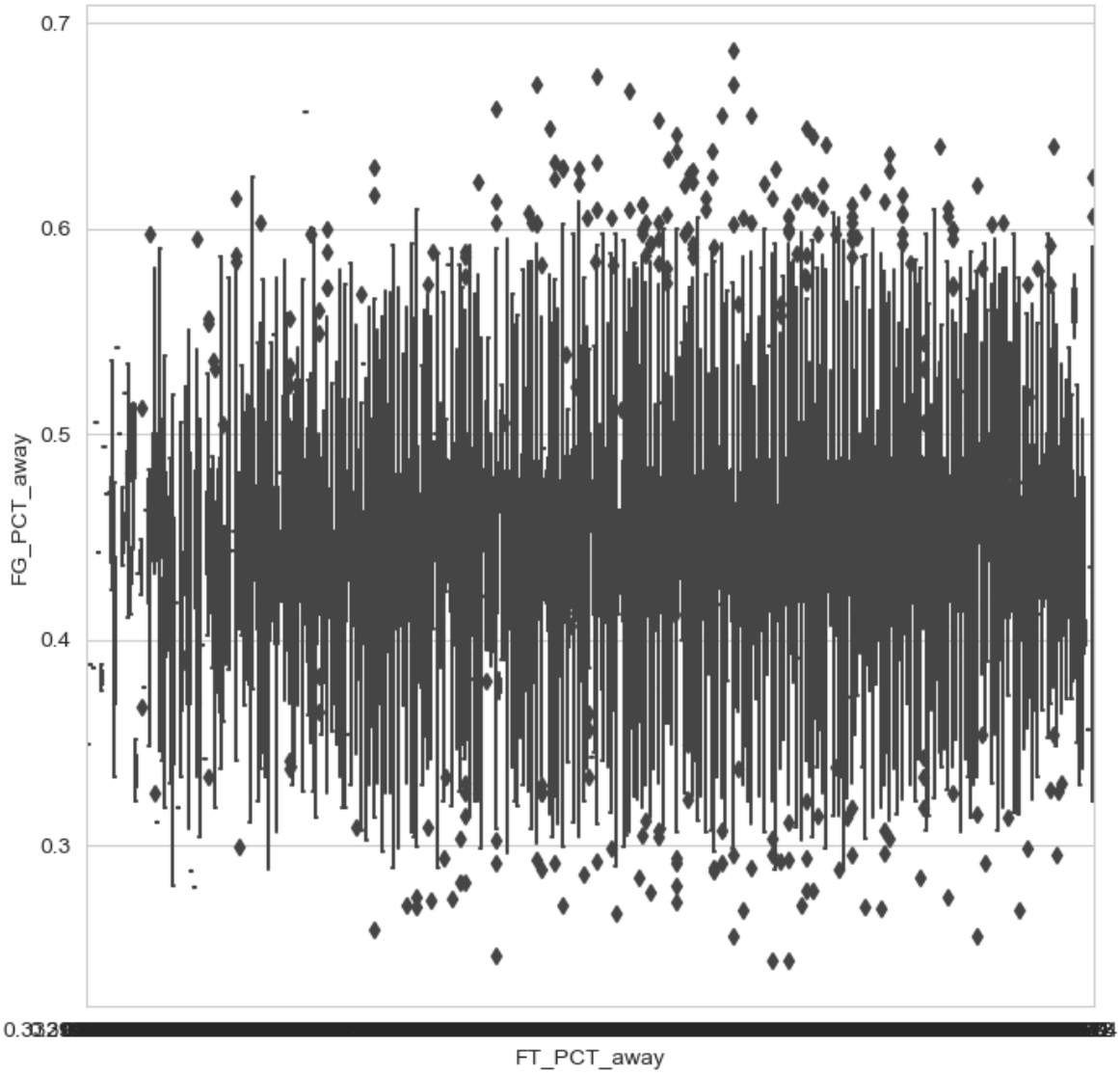


In [75]:
```
pip install plotly
```

```
Requirement already satisfied: plotly in c:\users\swethak\anaconda3\lib\site-packa
ges (5.9.0)
Requirement already satisfied: tenacity>=6.2.0 in c:\users\swethak\anaconda3\lib\s
ite-packages (from plotly) (8.0.1)
Note: you may need to restart the kernel to use updated packages.
```

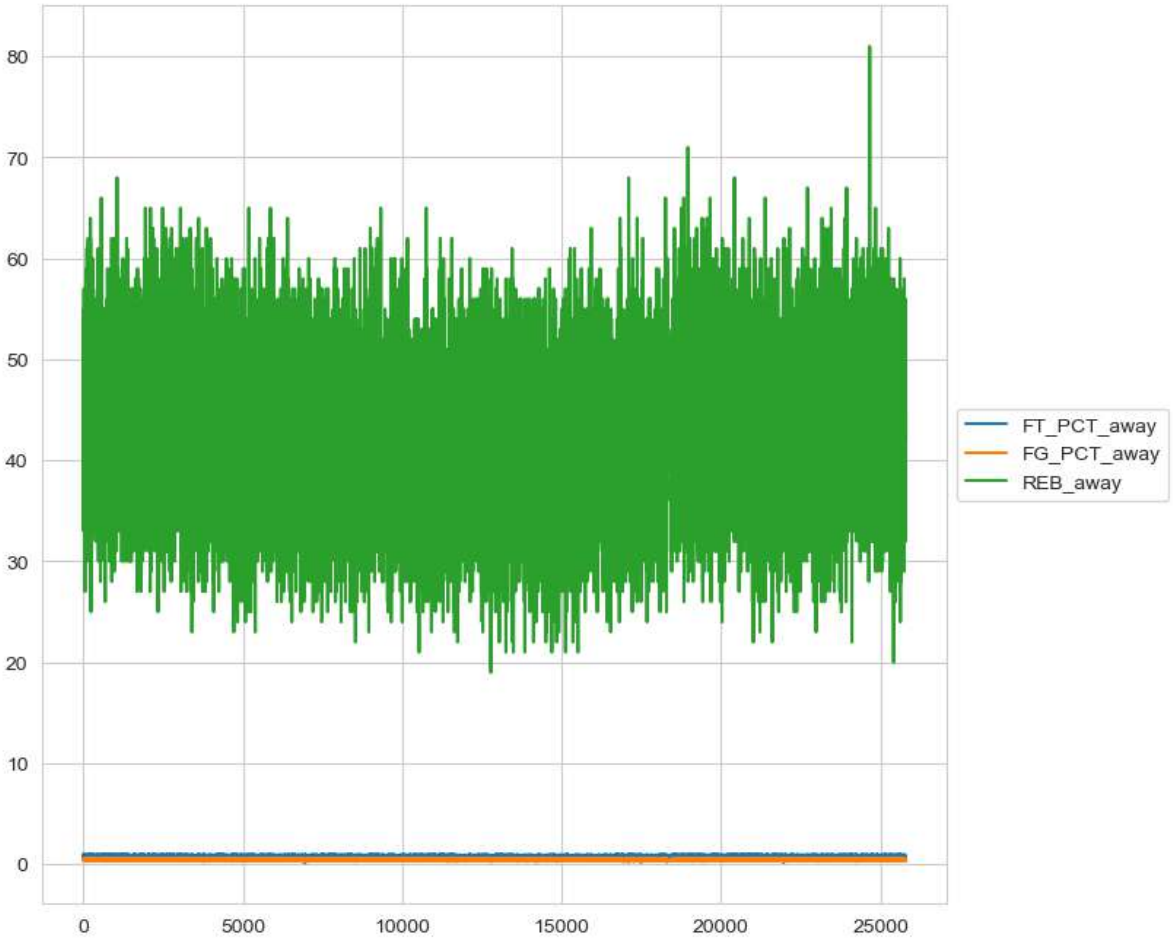
In [76]:
```
#importing libraries
import plotly.graph_objs as go
import plotly.offline as py
import plotly.express as px
from plotly.offline import iplot
```

In [77]:
```
fig, ax1 = plt.subplots(figsize=(8,8))
testPlot = sns.boxplot(ax=ax1, x='FT_PCT_away', y='FG_PCT_away', data=df)
```

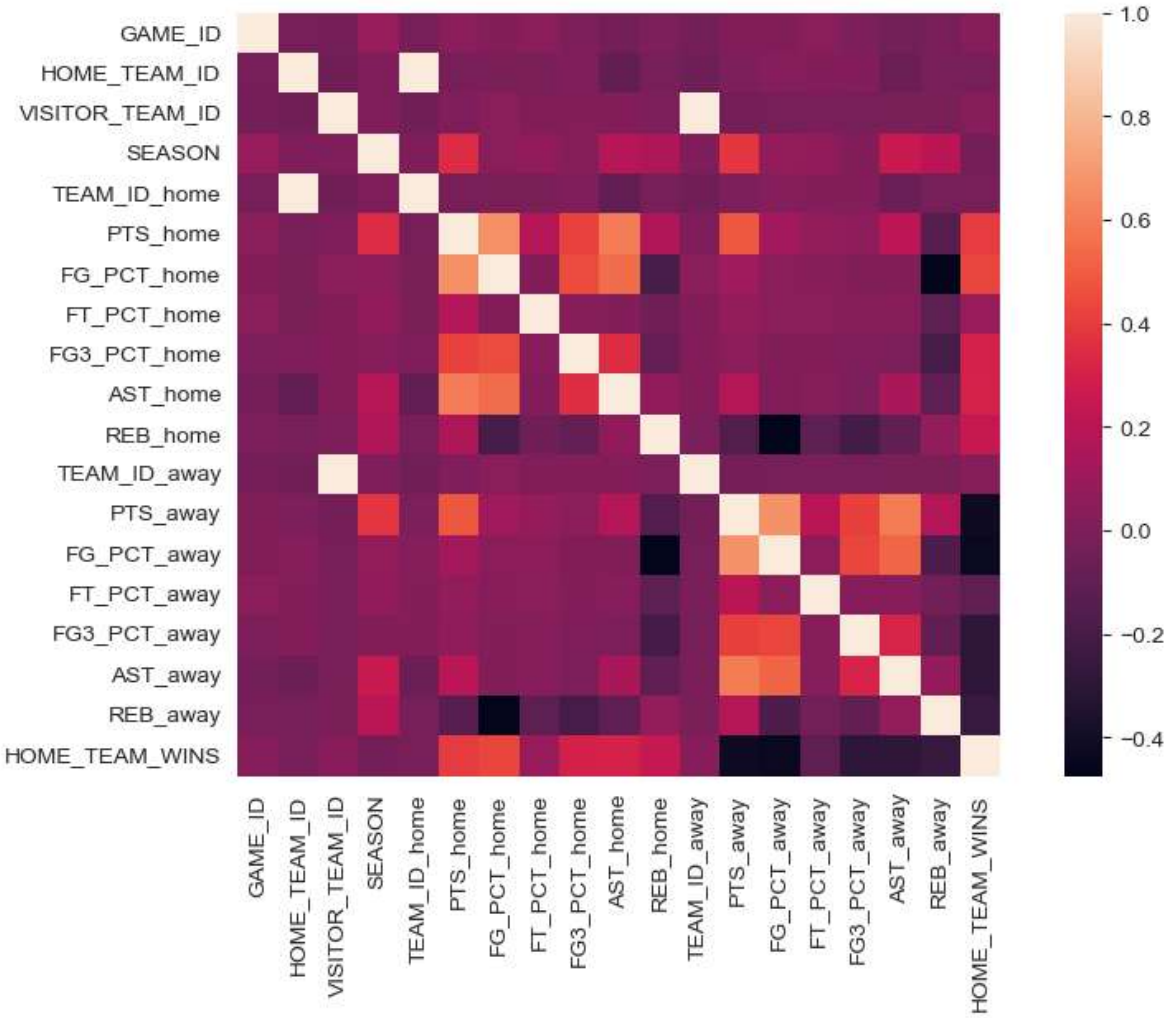


Profile plot

In [78]:
```
ax = df[["FT_PCT_away","FG_PCT_away","REB_away"]].plot(figsize=(8,8))
ax.legend(loc='center left', bbox_to_anchor=(1,0.5));
```

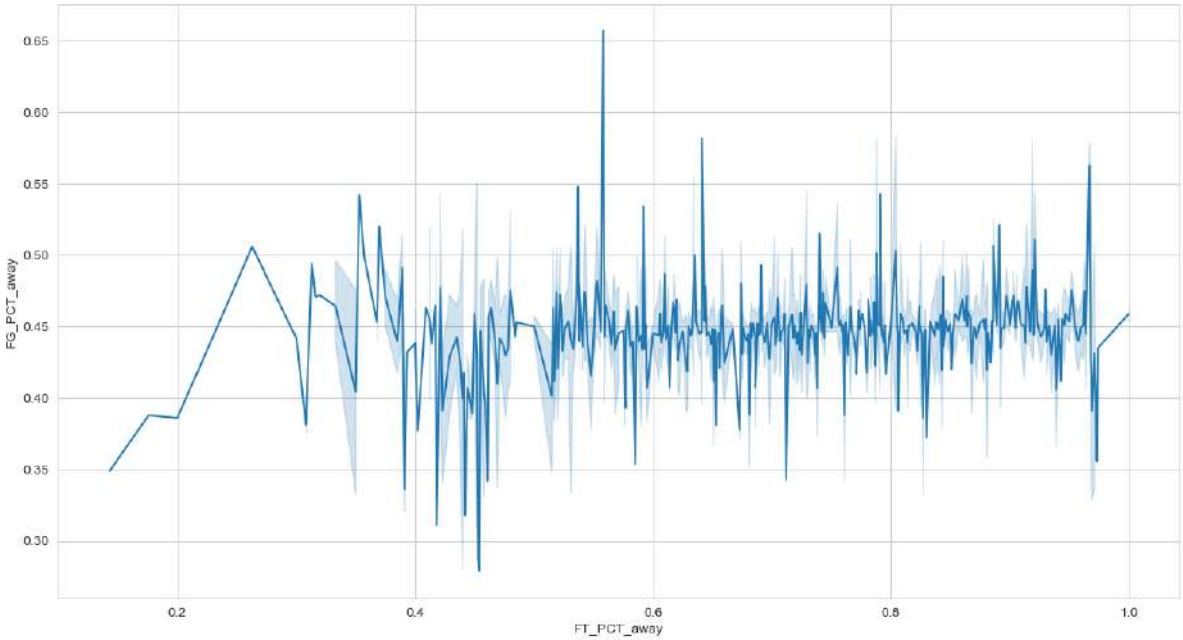


In [79]:
```
corr_df = df.corr()
plt.figure(figsize=(10,6))
sns.heatmap(corr_df,square=True)
plt.show()
```



In [80]:
```
plt.figure(figsize = (15,8))
fig = sns.lineplot( y ='FG_PCT_away',x = 'FT_PCT_away', data =df, palette = 'Set1'
fig.se
```

```
---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call last)
~\AppData\Local\Temp\ipykernel_3640\1673581078.py in <module>
      1 plt.figure(figsize = (15,8))
      2 fig = sns.lineplot( y ='FG_PCT_away',x = 'FT_PCT_away', data =df, palette
= 'Set1')
----> 3 fig.se

AttributeError: 'AxesSubplot' object has no attribute 'se'
```


0.65
0.60
0.55
0.50
0.45
0.40
0.35
0.30
FG_PCT_away
0.2
0.4
0.6
0.8
1.0
FT_PCT_away

In [ ]: